



Programación Paralela y Concurrente

1. Características generales

Nombre:	Programación Paralela y Concurrente
Sigla:	CI-0117
Créditos:	4
Horas lectivas:	5
Requisitos:	CI-0113 Programación 2, MA-0292 Álgebra Lineal para Computación, CI-0114 Fundamentos de Arquitectura
Correquisitos:	CI-0116 Análisis de Algoritmos y Estructuras de Datos
Clasificación:	Curso propio
Ciclo: de carrera	II ciclo, 2do año
Docente(s):	Gr01, Gr03: Jeisson Hidalgo-Céspedes Gr02, Gr04: Allan Berrocal Rojas
Grupo:	01, 02, 03, 04
Semestre y año:	II-2021
Horario y lugar de clases:	Gr01: L 9-11:50 / J 10-11:50 Zoom Gr02: K 9-11:50 / V 10-11:50 Zoom Gr03: L 13-15:50 / J 13-14:50 Zoom Gr04: K 13-15:50 / V 13-14:50 Zoom Todos: Aula virtual Telegram
Horario y lugar de consulta:	Prof. Jeisson: L 16-17 / K 9-12 / J 15-16 Zoom con cita previa Prof. Allan: L 13-17, J 13-17 Zoom con cita previa
Modalidad:	100% virtual

2. Descripción

La programación concurrente y distribuida de datos y tareas es una ampliación de las habilidades de programación serial desarrolladas en cursos previos. Esta ampliación es imprescindible por cuanto las plataformas de hardware actuales y futuras ofrecen características de computación paralela que no podrían ser aprovechadas por programadores que solo dominen la programación serial.

3. Objetivos

Objetivo general

El objetivo general del curso es que los estudiantes desarrollen habilidades para resolver problemas que requieren incremento del desempeño mediante la





paralelización de datos y tareas a través de los paradigmas de programación concurrente y distribuido.

Objetivos específicos

Durante el curso el estudiante desarrollará habilidades para:

1. Comprender y explicar las motivaciones y tendencias de la computación paralela para contextualizar el desarrollo de software paralelo en la actualidad mediante el estudio de las características generales de las tecnologías más relevantes.
2. Diseñar soluciones concurrentes y distribuidas correctas.
3. Resolver problemas por paralelización de datos y paralelización de tareas.
4. Construir programas concurrentes para resolver problemas mediante hilos de ejecución y recursos compartidos.
5. Construir programas distribuidos para resolver problemas mediante procesos y recursos distribuidos.
6. Probar programas concurrentes y distribuidos para asegurar su efectividad y calidad mediante el uso de pruebas de software.
7. Evaluar y comparar el desempeño de programas concurrentes y distribuidos para determinar su incremento en el rendimiento respecto a versiones funcionalmente equivalentes mediante la aplicación de métricas básicas de uso común.
8. Explicar el modelo de ejecución concurrente y distribuido (máquina nociónal) para implementar programas de forma correcta.

4. Contenidos

Objetivo	Eje temático	Desglose
1 Comprender motivaciones y tendencias	1.1 La necesidad de computación paralela	Las necesidades de separación de asuntos y desempeño que motiva el software paralelo.
	1.2 Hardware paralelo	Sinopsis de modelos de hardware paralelo. Jerarquía de Flynn
	1.3 Software paralelo	Concepto de proceso y de hilo de ejecución.
2 Diseñar soluciones	2.1 Algoritmos paralelos	Diferencia con algoritmos seriales. Análisis espaciotemporal de algoritmos paralelos.
	2.2 Técnicas de descomposición	Descomposición recursiva, de datos, exploratoria, especulativa y otras.





concurrentes y distribuidas	2.3 Mapeo de tareas a procesos	Características de tareas e interacciones. Técnicas de mapeo para balanceo de carga. Técnicas para reducir la sobrecarga debida a la interacción de tareas.
	2.4 Modelos de programas paralelos	Paralelismo de datos, grafo de tareas, <i>work-pool</i> y otros.
3. Resolver problemas	3.1 Problemas de paralelización de datos y tareas	Resolver problemas como: consumidor-productor, filósofos comensales, problema de los fumadores, problema de la barbería, problema de Santa Claus, formar agua, y el problema de <i>Modus Hall</i> .
4, 8 Construir programas concurrentes con recursos compartidos	4.1 Concurrencia por hilos	Concepto de hilo de ejecución. Espacio de direcciones. Interfaces de programación por hilos (como Pthreads y OpenMP). Rastreo de memoria y procesamiento de hilos de ejecución.
	4.2 Integridad de hilos	Condiciones de carrera (regiones críticas). Código reentrante. Código <i>thread-safe</i> .
	4.3 Mecanismos de sincronización	Espera activa. Mecanismos de sincronización provistos por el API (como <i>mutex</i> , semáforos, candados de lectura-escritura, variables de condición, barreras, reducciones).
5, 8 Construir programas concurrentes con recursos distribuidos	5.1 Concurrencia por procesos	Concepto de proceso. Memoria distribuida. Interfaces de programación por procesos (como MPI). Rastreo de memoria y procesamiento de procesos.
	5.2 Entrada y salida	Entrada y salida mediante procesos paralelos.
	5.3 Comunicación	Comunicación punto a punto y comunicación colectiva entre procesos.
6 Probar	6.1 Pruebas de software	Pruebas de correctitud en programas concurrentes y distribuidos (como caja negra y caja blanca).
7 Evaluar y comparar	7.1 Métricas	Ley de Amdahl. Métricas de aceleración (<i>speedup</i>), eficiencia, y escalabilidad.
	7.2 Desempeño	Medición del tiempo de pared. Gráficos de desempeño.

5. Metodología

Se sigue una metodología híbrida tradicional-constructivista 100% virtual. Las lecciones del curso pueden intercalarse entre lecciones magistrales sincrónicas a distancia y aula invertida.

En las lecciones magistrales, a través de la plataforma de videoconferencia, se utilizan recursos audiovisuales para ilustrar conceptos e implementaciones. Las lecciones serán grabadas, y se podrán hacer disponibles en línea para la consulta posterior de estudiantes en función de la disponibilidad de los docentes, así como otros recursos de referencia.





En el caso de aula invertida, los estudiantes realizan el estudio independiente de contenidos provistos por el docente, y las lecciones virtuales se dedicarán al acompañamiento de estudiantes en el desarrollo de ejercicios y el proyecto relacionados a dichos contenidos. Durante las horas extra-clase los alumnos estudiarán el material del curso y resolverán los ejercicios planteados.

La consulta será a través de videoconferencia mediante la herramienta de indicada por los docentes. Se requiere cita previa, la cual se puede obtener por mensajería instantánea o correo electrónico. Todas las consultas con los docentes serán grabadas. Se provee además un grupo de mensajería instantánea para consultas de interés de todos los estudiantes.

6. Evaluación

%	Rubro	Descripción
25 %	Tareas	Las tareas se asignarán en Mediación Virtual u otras herramientas digitales indicadas por los docentes. Pueden ser de ponderaciones variables asignadas a criterio de los docentes. Los medios para entregar las soluciones serán indicados en los respectivos enunciados.
25 %	Quices	Los docentes podrán realizar exámenes cortos (quices) en cualquiera de las clases magistrales y al momento de la clase que los docentes lo consideren apropiado. Estos se realizarán en Mediación Virtual. Los quices serán de igual ponderación y la duración de cada quiz será especificada en su enunciado. De acuerdo con el artículo 15 del Reglamento de Régimen Académico Estudiantil dichas reglas quedan establecidas después de este aviso.
30 %	Proyecto 1	Proyecto orientado a concurrencia y distribución de tareas. Se evalúa con al menos tres entregables de igual ponderación, tentativamente en las semanas 6, 10, y 16.
20 %	Proyecto 2	Proyecto orientado a concurrencia y distribución de datos. Se evalúa con al menos dos entregables de igual ponderación, tentativamente en las semanas 8 y 14.

Los proyectos serán elaborados por equipos de tres o cuatro estudiantes. Los avances de los proyectos se evalúan mediante videoconferencia entre los docentes y los miembros del equipo quienes realizan una presentación oral. Las fechas de presentación de avances y rubros de evaluación serán especificados en sus respectivos enunciados, no así el contexto del problema como se indica a continuación. Los requerimientos del dominio de cada proyecto serán especificados oralmente por los docentes. Los estudiantes elaborarán el análisis del problema en un documento de requerimientos, un diseño de su solución, además de la implementación en software de la solución y las rutinas de prueba. Este esquema pretende emular la práctica que ocurre en la industria, donde los





clientes especifican verbalmente sus necesidades y los profesionales en informática recaban los requerimientos.

Lineamientos de evaluación:

1. Toda asignación se considera tardía si se entrega después de las 6 a.m. posterior a la fecha de entrega. Si la entrega tiene 24 o menos horas de retraso se le aplicará una penalización de 50% del valor. Después de 24 horas de retraso, la asignación será calificada con cero.
2. Las tareas o avances de proyecto deberán ser entregados al profesor el día propuesto en el enunciado, por los medios indicados en el enunciado de la evaluación.
3. Los quices se realizan durante las lecciones y en cualquier momento durante el transcurso de la lección, y solo se repondrán en los casos que establece el Reglamento de Régimen Académico Estudiantil en su artículo 24. Se recomienda a los estudiantes contar con medios adicionales que garanticen el acceso a Mediación Virtual en caso de anomalías como fallos eléctricos o de telefonía fija, como disponer de la aplicación Moodle o un navegador para dispositivos móviles pre-configurados.
4. Todas las evaluaciones son estrictamente individuales excepto en aquellas en las que se especifique explícitamente el trabajo grupal en el enunciado.
5. Todo trabajo debe ser entregado de forma digital.
6. En todos los trabajos y las evaluaciones de los estudiantes, se calificará la redacción, ortografía, estructura y contenido.
7. Todo código fuente que no compile será calificado con cero.
8. En toda asignación se evaluarán las buenas prácticas de programación, como identificadores significativos, indentación, apego a una convención de estilo, documentación de interfaces e implementaciones de subrutinas, modularización y reutilización de código. Serán castigadas malas prácticas de programación como redundancia de código, fugas y accesos inválidos de memoria, condiciones de carrera, serialización innecesaria de la concurrencia, y otras prácticas estipuladas durante las lecciones del curso.
9. Si se detecta que una solución contiene una espera activa, será calificada con cero.
10. Los docentes podrían ofrecer un incremento en la nota hasta un máximo de la mitad de la diferencia por correcciones tras una evaluación revisada. Este es un beneficio que los docentes pueden ofrecer a su discreción en los enunciados de las evaluaciones. Para los proyectos este beneficio se pierde en caso de que la persona estudiante realice un cambio de equipo de trabajo.





11. En cualquier evaluación se podrían ofrecer rubros opcionales por crédito extra a discreción de los docentes.

Es ilegal presentar como propio, código parcial o total escrito por otras personas u obtenido de fuentes de información, como por ejemplo de libros o de Internet, sin la debida referencia al autor de la propiedad intelectual. En cualquier asignación en que se sospeche de plagio se aplicará el debido proceso estipulado en el [Reglamento de Orden y Disciplina de los Estudiantes de la Universidad de Costa Rica](#).

7. Cronograma

La tabla de abajo muestra las fechas tentativas de cobertura de los contenidos, que pueden ser reajustadas de acuerdo con el avance durante el ciclo lectivo. Las fechas de entrega de los ejercicios y avances de proyecto estarán sujetas a esta cobertura temática, y se comunicará oportunamente. Los temas se van a cubrir en el orden en que se presentan en la siguiente tabla.

#	Temática	Semanas
1	Introducción a la concurrencia y distribución Ejes temáticos: 1.1 a 1.3	1
2	Concurrencia compartida imperativa (Pthreads) Ejes temáticos: 2.1 a 2.4, 3.1, 4.1 a 4.3, 6.1, 7.1, 7.2 Problemas de control de concurrencia (eje temático 3.1)	10
3	Concurrencia compartida declarativa (OpenMP) Ejes temáticos: 2.1 a 2.3, 4.1 a 4.3, 6.1, 7.1, 7.2	2
4	Concurrencia distribuida (MPI) Ejes temáticos: 2.2, 2.3, 5.1 a 5.3, 6.1, 7.1, 7.2	3

8. Bibliografía

1. Pacheco, Peter S. "An introduction to parallel programming". Morgan Kaufmann Pub, 2011.
2. Rauber, Thomas y Rünger, Gudula. "Parallel Programming (for multicore and cluster systems)". Springer-Verlag Berlin Heidelberg, 2010.
3. Grama, Ananth et.al. "Introduction to Parallel Computing". Addison-Wesley, 2003.
4. Quinn, Michael J. "Parallel Programming in C with MPI and OpenMP". McGraw-Hill Education, 2003.
5. Chandra, Rohit et.al. "Parallel Programming in OpenMP". Morgan Kaufmann Pub, 2001.
6. Downey, Allen B. Little Book of Semaphores.
Disponible en <https://greenteapress.com/wp/semaphores>
7. Zaconne, Giancarlo. "Python Parallel Programming" 2nd Ed., 2019
8. Trobec, Roman; Slivnik, Boštjan; Bulić, Patricio & Robič, Borut.
"Introduction to Parallel Computing (From Algorithms to Programming on State-of-the-Art Platforms)". Springer, 2018





9. Peterson, James Lyle. "Petri Net Theory and the Modeling of Systems". Prentice Hall PTR, USA. 1981
10. Reisig, Wolfgang. [Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies](#). Springer Publishing Company, Incorporated. 2013

9. Recursos estudiantiles

El Sistema de Bibliotecas, Documentación e Información (SIBDI) de la Universidad de Costa Rica cuenta con una amplia gama de recursos de información bibliográfica en diferentes formatos como libros, folletos, publicaciones periódicas, trabajos finales de graduación, entre otros. Desde la [Biblioteca virtual](#) se puede acceder a muchos de estos recursos, incluyendo publicaciones en conferencias y revistas del área de Computación (indexadas por las editoriales [ACM](#), [IEEE](#), y [ScienceDirect](#), entre otras) y colecciones de libros electrónicos como [eLibro](#) y [AccessEngineering](#). La Biblioteca Luis Demetrio Tinoco ofrece cursos de capacitación para estudiantes del área Ingeniería y Computación.

El [sitio web](#) del Consejo Universitario de la UCR contiene las diferentes [normativas estudiantiles](#) que rigen en la UCR. En particular, los procedimientos de evaluación y orientación establecidos en el Reglamento de Régimen Académico Estudiantil se pueden consultar en [este enlace](#). De dicho reglamento, destacamos especialmente los siguientes artículos (que recomendamos leer y estudiar):

- El artículo 14 se refiere al contenido de los programas de los cursos.
- El artículo 17 indica en qué circunstancias se pueden variar las normas de evaluación de un curso.
- El artículo 22 establece el procedimiento en relación con la calificación, entrega e impugnación de los resultados de cualquier prueba de evaluación.
- El artículo 24 establece el procedimiento para solicitar la reposición de evaluaciones.

