



Arquitectura de Computadoras

1. Características generales

Nombre:	Arquitectura de Computadoras
Sigla:	CI-0120
Créditos:	4
Horas lectivas:	5 horas de teoría
Requisitos:	CI-0114 Fundamentos de Arquitectura, CI-0118 Lenguaje Ensamblador
Correquisitos:	Ninguno
Clasificación:	Curso propio
Ciclo de carrera:	I ciclo, 3er año (énfasis de Ciencias de la Computación)
Docente(s):	Francisco Arroyo Mora
Datos de contacto:	Oficina 240, rafael.arroyo@ucr.ac.cr , francisco.arroyo@eccci.ucr.ac.cr
Grupo:	Grupo 1
Semestre y año:	I ciclo 2021
Horario y lugar de clases:	K 10, 11 y V 9, 10, 11 Nota: modalidad 100% virtual, las clases serán desarrolladas en modo sincrónico por medio de Zoom, a menos que se indique de otra manera
Horario y lugar de consulta:	M, J 13, 14, 15, 16 por medio de herramientas de intercambio de mensajes o en reunión virtual agendada con cita previa
Asistente:	Por definir

2. Descripción

Este curso está dirigido a estudiantes que inician su tercer año de bachillerato. Se enfoca en el estudio de distintas arquitecturas de computadoras para permitirle al estudiante tomar decisiones acertadas sobre las capacidades de un computador para ejecutar un determinado programa de manera eficiente, especialmente en ambientes de ejecución paralela. Se requieren conocimientos amplios de programación en C/C++, ensamblador y lenguaje de máquina. Durante el curso, el estudiante adquiere un mayor



criterio que le permite evaluar y comparar el rendimiento de diferentes arquitecturas de computadoras. El curso puede darse bajo un enfoque por proyectos o mediante una combinación de clases magistrales, estudio independiente y tareas con el uso de laboratorios.

3. Objetivos

Objetivo General

El objetivo general de este curso es que cada estudiante comprenda los principios de diseño que rigen las principales arquitecturas de computadoras para determinar su impacto en el rendimiento de los programas.

Objetivos Específicos

Durante este curso cada estudiante desarrollará habilidades para:

1. Relacionar la arquitectura de una computadora con su conjunto de instrucciones.
2. Utilizar técnicas que permitan obtener estadísticas dinámicas sobre los tipos de instrucciones ejecutadas.
3. Realizar y evaluar diferentes mediciones de desempeño de procesadores con el fin de comprender cómo y por qué la arquitectura del procesador en el que se trabaja es un elemento determinante en el comportamiento de las aplicaciones de software que ahí se ejecuten.
4. Analizar críticamente un microprocesador calculando medidas de rendimiento con base en sus elementos y la organización de su arquitectura.
5. Comparar y contrastar distintas organizaciones de memoria.
6. Aplicar técnicas que aprovechen el paralelismo inherente a un proceso.
7. Explicar las diferencias que existen entre distintos procesadores y multiprocesadores con implementaciones de paralelismo de hilos, de procesos y de datos.



4. Contenidos

Objetivo específico	Eje Temático	Desglose
1 y 2	Fundamentos de diseño	El diseño de computadoras. Clases de computadoras. Definición de la arquitectura de una computadora. Tendencias en la tecnología de circuitos integrados y sus costos, consumo de energía y eficiencia. Medición del desempeño. Principios cuantitativos en el diseño de computadores.
1 y 3	Arquitectura basada en el conjunto de instrucciones	Clasificación. Direccionamiento de memoria. Modos de direccionamiento para procesamiento de señales. Tipo y tamaño de los operandos. Operandos para procesamiento de señales y multimedia. Operaciones para procesamiento de señales y multimedia. Instrucciones para el control de flujo. Codificación de un conjunto de instrucciones. Rol de los compiladores. Relación con lenguajes de alto nivel. Arquitectura MIPS.
1, 6 y 7	Paralelismo a nivel de instrucciones y ejecución segmentada (<i>pipeline</i>)	Definición de <i>pipelining</i> . Problemas con el pipeline: conflictos de datos y de control. Dificultades para implementar <i>pipelining</i> (excepciones). Pipeline para operaciones multiciclo y ejemplo del <i>pipeline</i> para el procesador MIPS. Procesadores superescalares (de emisión múltiple).
5	Jerarquías de memoria	Introducción a las memorias caché. <i>Translation lookaside buffers</i> (TLB). Tecnología de memoria y optimizaciones. Desempeño de la caché. Reducción de tasa de fallos de caché. Diseño de jerarquías de memoria y memoria virtual. Protección y ejemplos de memoria virtual.



4, 6 y 7	Paralelismo a nivel de datos	Arquitecturas vectoriales, SIMD (<i>single instruction, multiple data</i>) y GPU (<i>graphics processing unit</i>). Extensiones SIMD del conjunto de instrucciones para multimedios.
6 y 7	Paralelismo a nivel de hilos	Arquitecturas de memoria compartida centralizada. Multiprocesadores simétricos (SMP), Acceso uniforme a memoria (UMA). Desempeño de los multiprocesadores de memoria compartida simétrica. Coherencia de caché mediante protocolos <i>snooping</i> . Arquitecturas de memoria compartida distribuida no uniforme (NUMA). Coherencia de caché mediante protocolos de directorios.
5	Almacenamiento externo	Discos duros, memorias <i>flash</i> y discos de estado sólido (SSD).
3	Redes de interconexión y clusters	Creación de procesos en ambientes de multiprocesadores.

5. Metodología

El curso será 100% virtual, su dinámica de virtualización será seguir reuniones sincrónicas no presenciales con herramientas especializadas en el horario establecido para cada grupo, en las que el profesor presentará el material correspondiente a la temática, realizará actividades prácticas relacionadas y conformará grupos para resolución de problemas determinados. Se utilizará la plataforma Mediación Virtual (<https://mv1.mediacionvirtual.ucr.ac.cr/>).

Además, dispondremos de grupos para intercambio de mensajes, *Telegram* o similar. En el caso de las evaluaciones, ya sean exposiciones, exámenes cortos o exámenes, los estudiantes deberán obligatoriamente estar conectados en tiempo real, con una cámara encendida. Para las clases normales o bien otras actividades este requerimiento es opcional.



6. Evaluación

Item	Evaluación	Porcentaje	Fecha
1	Examen parcial	25	Vierne 28 de Mayo, 1 p.m.
2	Examen parcial	25	Viernes 30 de Julio, 1 p.m.
3	Tareas y exámenes cortos	25	
4	Proyecto final	25	

Notas:

- Cada evaluación indicará en su enunciado las normas de evaluación
- Los exámenes cortos se harán en horario de clases y comprenderán la materia vista más reciente, se trata de asignaciones individuales
- Las tareas cortas tendrán un plazo de entrega indicado en cada una de ellas, se trata de asignaciones individuales
- Es obligatorio que cada estudiante lea el reglamento de Régimen Académico Estudiantil, ver el enlace en la sección 9
- Todos los trabajos deben ser entregados en las fechas acordadas, en formato impreso o digital según se indique. La penalidad por entrega tardía es de 10 pts. (de 100) por día natural, con un plazo de no más de tres días naturales de atraso, pasado este tiempo la asignación no será calificada. Los puntos son rebajados de la nota obtenida, siendo el máximo rebajo 30 pts. Se considera un trabajo entregado a partir de la recepción en la plataforma o en el correo electrónico indicado
- En caso de descubrir plagio o copiado de los trabajos, se pondrá un cero en la asignación y aplicarán las sanciones establecidas en la normativa universitaria de la UCR: REGLAMENTO DE ORDEN Y DISCIPLINA DE LOS ESTUDIANTES. UCR. ARTÍCULO 4. Son faltas muy graves
 - ...
 - j) Plagiar, en todo o en parte, obras intelectuales de cualquier tipo
 - k) Presentar como propia una obra intelectual elaborada por otra u otras personas, para cumplir con los requisitos de cursos, trabajos finales de graduación
- CAPÍTULO III DE LAS SANCIONES ARTÍCULO 9. Las faltas serán sancionadas según la magnitud del hecho con las siguientes medidas:
 - Las faltas muy graves, con suspensión de su condición de estudiante regular no menor de seis meses calendario, hasta por seis años calendario



7. Cronograma

Será detallado en Mediación Virtual.

8. Bibliografía

- [1] **John L. Hennessy y David A. Patterson**, *Computer Architecture: A Quantitative Approach*. Elsevier Inc. / Morgan Kaufmann Publishers, San Francisco, 6ª edición (2019) [**Texto**]
- [2] **William Stallings**, *Computer Organization and Architecture*, Pearson, 10ª edición (2015).
- [3] **David A. Patterson y John L. Hennessy**, *Computer Organization and Design*, Elsevier Inc. / Morgan Kaufmann Publishers, San Francisco, 5.a edición (2014).



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](#), ente 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):

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

