Lenguaje Ensamblador

1. Características generales

Nombre: Lenguaje Ensamblador

Sigla: CI-0118

Créditos: 4

Horas lectivas: 5 horas de teoría

Requisitos: CI-0114 Fundamentos de Arquitectura

Correquisitos: CI-0119 Proyecto Integrador de Arquitectura y Ensamblador

Clasificación: Curso propio

Ciclo de carrera: II ciclo, 2do. año

Docente: Dr. Carlos Vargas

Datos de contacto: Oficina 232, carlos.vargas@ucr.ac.cr, 2511-8006

Semestre y año: I ciclo 2025

Grupo: **01**

Modalidad: Curso Regular

Grado de Virtualidad: Bajo virtual

Se utiliza Mediación Virtual como apoyo para programación de eventos, entrega de materiales y calificación de pruebas.

Horario y lugar de clases: K 13 A 15:50/ V 13 A 14:50, aula 203IF

Horario y lugar de consulta: V 15:00 a 17:30 presencial y adicionalmente por correo electrónico o "Zoom", con cita previa.

2. Descripción

Este curso toma como base conceptos de arquitectura de la CPU y su relación con los dispositivos periféricos para aprender el lenguaje ensamblador y comprender su relación con los lenguajes de alto nivel. Además, se estudian entornos especializados de aplicación del lenguaje ensamblador.

3. Objetivos

Objetivo general

Utilizar el lenguaje ensamblador y comprender su relación con los lenguajes de alto nivel y el sistema operativo, para hacer un uso eficiente y eficaz de los recursos de la computadora haciendo énfasis en esa relación, mediante el estudio de un lenguaje ensamblador y su aplicación en la solución de problemas.

Objetivos específicos

Durante este curso cada estudiante desarrollará habilidades para:

1. Analizar las capacidades y limitaciones de una arquitectura para comprender su relación con los lenguajes de alto nivel y el sistema operativo mediante el estudio de la estructura lenguaje ensamblador.
2. Diseñar e implementar programas para hacer un uso eficiente y eficaz de los recursos de la computadora utilizando lenguaje ensamblador.
3. Desarrollar programas en lenguaje ensamblador que se comuniquen con dispositivos periféricos utilizando servicios de bajo nivel del sistema operativo para comprender los mecanismos de interacción de la computadora con el entorno externo.
4. Diseñar e implementar programas que integren el lenguaje ensamblador con lenguajes de alto nivel para resolver problemas de forma más eficiente o eficaz utilizando lenguajes de bajo nivel.
5. Resolver problemas clásicos de bajo nivel mediante el desarrollo de aplicaciones específicas de lenguaje ensamblador para dar una visión global del rol del lenguaje ensamblador en el ecosistema computacional.

Transversales

Además, cada estudiante desarrollará habilidades en los siguientes ejes trasversales:

1. Seguridad
2. Trabajo en equipo
3. Computación paralela y distribuida
4. Buenas prácticas de programación
5. Optimización

4. Contenidos

Objetivos	Eje temático	Desglose
1	Conceptos básicos	• Historia del lenguaje máquina y compiladores • Otras arquitecturas (*RISC*, máquinas de pila, etc.) • Instrucciones del lenguaje ensamblador • Macros y directivas del ensamblador • Ensamblaje y desensamblaje de instrucciones • Ingeniería reversa • Compilador y depuración • *Linking*, *loader*, relocalización, resolución de símbolos
1, 2	Programación en lenguaje ensamblador	• Representación de datos (enteros – *shortint*, *longint* –, cadenas, punto flotante, *ascii*, *ansi*, *unicode*) • *Endianness* (*little endian*, *big endian*) • *Bitness* (compatibilidad entre arquitecturas de software – ej. 32 vs. 64 bits) • Modos de direccionamiento de la memoria (relación con el direccionamiento en alto nivel) • Optimización del uso de la memoria y de cachés
2, 3	Comunicación con dispositivos periféricos	• Interrupciones, excepciones, procesos, señales • Programación del sistema de vídeo (GPU) • Programación con funciones del BIOS (arranque, entrada/salida)
4	Relación con lenguajes de alto nivel	• Interfaz con lenguajes de alto nivel • *Stack frame* • Recursividad • Prólogo • Epílogo • Paso de parámetros • Convenciones de uso de registros • Alcance de las variables
3, 4, 5	Problemas conocidos de bajo nivel	• Coprocesador matemático y punto flotante en el procesador • Optimización y comparación • *Device drivers* • Fundamentos del *boot manager* • Graficación • *Buffer overflow* • Soporte a virtualización

5. Metodología

El curso "Lenguaje Ensamblador" ofrece al estudiante los conocimientos teóricos y prácticos para comprender el acople entre la arquitectura de una computadora (hardware) y los programas que ejecuta (software) y para desarrollar programas óptimos en ensamblador.
La modalidad de este curso es fundamentalmente presencial, incluyendo un bajo uso de aula virtual. Este curso se imparte con una fuerte orientación pedagógica de "aprendizaje orientado a proyectos", donde el estudiante tiene un papel activo como generador de conocimiento. A su vez, el docente facilita los procesos de enseñanza y aprendizaje, incluyendo espacios adecuados que potencian el trabajo autónomo en equipo. Estos incluyen una sesión asincrónica y otra sincrónica (vía Zoom) desarrollados bajo un enfoque pedagógico de "Aula Invertida". Además de exámenes, el curso incluye investigación, tareas programadas, tareas cortas y quices. Se utiliza Mediación Virtual (https://mediacionvirtual.ucr.ac.cr/) como apoyo para programación de eventos, entrega de materiales y calificación de pruebas. Consulta presencial y por medio de reuniones "Zoom", con cita previa.

6. Evaluación

Los 100 puntos, correspondientes a la máxima calificación que se puede obtener en el curso, se distribuyen de la siguiente forma:

Ítem	**Porcentaje de la nota**
Examen Parcial I	20%
Examen Parcial II	20%
Quices y Tareas Cortas	10%
Proyecto de Investigación	20%
Dos tareas programadas	30% (15% cada una)

Para aprobar el curso el estudiante debe obtener al menos 67.5 puntos al finalizar el semestre. Si la nota final está entre 57.5 y 67.4 tendrá derecho a realizar un examen de ampliación. En este examen el estudiante deberá obtener una nota mínima de 70.0 para aprobar el curso. En caso de que obtenga una nota menor a 57.5, o de presentar el examen de ampliación con una nota inferior a 70.0 reprobará el curso.

7. Cronograma

Desarrollo de los contenidos

Objetivos	Eje temático	Duración
1	Conceptos básicos	3 semanas
1, 2	Programación en lenguaje ensamblador	7 semanas
2, 3	Comunicación con dispositivos periféricos	2 semanas
4	Relación con lenguajes de alto nivel	2 semanas
3, 4, 5	Problemas conocidos de bajo nivel	2 semanas

Fechas de entrega de tareas programadas

- Tarea programada No. 1: 29 de mayo *de* 2025.
- Tarea Programada No. 2: 24 de junio de 2025.

Fechas de Exámenes

- Examen Parcial I: 27 de mayo del 2025, 1 PM.
- Examen Parcial II: 17 de junio del 2025, 1 PM.

Entrega reporte de investigación: 29 de abril del 2025.

8. Bibliografía

Abel, Peter. IBM PC Assembly Language and Programming. Pearson Education. Quinta Edición. 2001.

Brey, Barry. The Intel Mircroprocessors. Pearson Education. Octava Edición. 2011.

Bryant, Randal y O'Hallaron, David. Computer Systems, A programmer's perspective. Tercera Edición. 2015.

Irvine, Kip. Assembly Language for x86 Processors. Pearson Educación. Octava Edición. 2019.

Intel. Intel® 64 and IA-32 Architectures Software Developer's Manual Combined Volumes: 1, 2A, 2B, 2C, 2D, 3A, 3B, 3C, 3D, and 4 (https://software.intel.com/en-us/download/intel-64-and-ia-32-architectures-sdm-combined-volumes-1-2a-2b-2c-2d-3a-3b-3c-3d-and-4). 2016.

Jorgensen, Ed. x86-64 Assembly Language Programming with Ubuntu (http://www.egr.unlv.edu/~ed/assembly64.pdf). 2020.

9. Recursos estudiantiles y normativa

Para información sobre recursos estudiantiles disponibles en la UCR, incluyendo el Sistema de bibliotecas y la normativa universitaria vigente, favor visitar la página :

https://www.ecci.ucr.ac.cr/vida-estudiantil/servicios-institucionales-para-estudiantes/guia-de-recursos-estudiantiles-de-la-ucr