



ORGANIZACIÓN DE LENGUAJES Y COMPILADORES 2
 Segundo Semestre 2024

I. INFORMACIÓN GENERAL

Código: 781	Créditos: 5
Escuela: Ciencias y Sistemas	Área: Ciencias de la Computación
El curso tiene laboratorio: Si	Categoría: Obligatorio
Horas magistrales a la semana: 10	Horas de laboratorio a la semana: 10
Prerrequisitos: 772 – Estructuras de datos 777 – Organización de lenguajes y compiladores 1	
Postrequisitos: 281 – Sistemas operativos 1	

II. DISTRIBUCIÓN DE SECCIONES

Sección	Edificio	Salón	De:	A:	Lu	Ma	Mi	Ju	Vi	Sa	Do	Catedrático
A	T-3	401	7:10	8:50	X							BAYRON WOSVELY LOPEZ LOPEZ
A	T-3	401	12:10	13:50						x		BAYRON WOSVELY LOPEZ LOPEZ
B	T-3	401	7:10	10:30		X						LUIS FERNANDO ESPINO BARRIOS
N	T-3	216	19:00	20:40		X		X				LUIS FERNANDO ESPINO BARRIOS

III. DESCRIPCIÓN DEL CURSO

Este curso es la continuación del estudio de las fases de un Compilador, específicamente el análisis de semántica y la fase de síntesis. Se tratan con detalle las definiciones dirigidas por la sintaxis, el manejo de la tabla de símbolos, la generación de código intermedio y optimización de código.

Se desarrollarán dos proyectos para aplicar los conceptos generales de compiladores, usando herramientas básicas tales como generadores de analizadores de léxico y de sintaxis.

IV. COMPETENCIA GENERAL

Que el estudiante desarrolle los conceptos básicos de las fases de un compilador para su diseño e implementación de tanto de compiladores como de intérpretes de un lenguaje de alto de nivel.

V. METODOLOGÍA

- Forma: virtual
- Método: deductivo
- Técnicas: expositiva y demostrativa
- Instrumentos: guías de trabajo, hojas de trabajo, ejercicios, preguntas, diálogo y observación
- Las clases magistrales se impartirán en 10 períodos semanales
- El laboratorio se impartirá 10 períodos semanales
- Durante el curso, se asignan 2 proyectos de laboratorio a realizarse de manera individual; así como tareas, ejercicios, prácticas e investigaciones.

VI. CALENDARIZACIÓN

	Junio																																				
Unidad 1	1	2	3	4	7																																
Parcial 1						8																															
Unidad 2							9	10	11	14	15																										
Parcial 2												16																									
Asueto													17																								
Unidad 3														18	21	22	23	24																			
Asueto																												25									
Parcial 3																																			28		
Final																																			29		
Asueto																																				30	

VII. CONTENIDO DECLARATIVO

UNIDAD 1: TRADUCCIÓN DIRIGIDA POR LA SINTAXIS
(10 PERIODOS)

Competencia específica: define los conceptos de la traducción dirigida por la sintaxis tanto en analizados sintácticos ascendentes como descendentes.

1. Traducción dirigida por la sintaxis
 - 1.1. Definiciones dirigidas por la sintaxis
 - 1.1.1. Atributos heredados y sintetizados
 - 1.1.2. Evaluación de una definición dirigida por la sintaxis en los nodos de un árbol sintáctico
 - 1.2. Órdenes de evaluación para las definiciones dirigidas por la sintaxis
 - 1.2.1. Gráficos de dependencias
 - 1.2.2. Orden de evaluación
 - 1.2.3. Definiciones con atributos sintetizados
 - 1.2.4. Definiciones con atributos heredados
 - 1.3. Aplicaciones de la traducción orientada por la sintaxis
 - 1.3.1. Construcción de árboles de análisis sintáctico
 - 1.3.2. La estructura de tipos
 - 1.4. Esquemas de traducción orientados por la sintaxis
 - 1.4.1. Esquemas de traducción postfijos
 - 1.4.2. Implementación de esquemas de traducción orientados a la sintaxis postfijo con la pila
 - 1.4.3. Esquema de traducción orientados a la sintaxis con acciones dentro de producciones
 - 1.4.4. Eliminación de la recursividad por la izquierda de los esquemas de traducción
 - 1.4.5. Esquemas de traducción orientados a la sintaxis para definiciones con atributos heredados por la izquierda
 - 1.5. Implementación de definiciones dirigidas por la sintaxis con atributos heredados por la izquierda
 - 1.5.1. Traducción durante el análisis sintáctico de descenso recursivo
 - 1.5.2. Generación de código al instante
 - 1.5.3. Las definiciones dirigidas por la sintaxis con atributos heredados por la izquierda y análisis sintáctico LL
 - 1.5.4. Análisis sintáctico ascendente de las definiciones dirigidas por la sintaxis con atributos heredados por la izquierda

UNIDAD 2: GENERACIÓN DE CÓDIGO INTERMEDIO (10 PERIODOS)

Competencia específica: genera código de tres direcciones a partir de los árboles de análisis sintáctico como parte de la fase de síntesis.

2. Generación de código intermedio
 - 2.1. Variantes de los árboles sintácticos
 - 2.1.1. Grafo dirigido acíclico para expresiones
 - 2.1.2. Método número de valor para GDA
 - 2.2. Código de tres direcciones

- 2.2.1. Direcciones e instrucciones
- 2.2.2. Cuádruplos
- 2.2.3. Tripletas
- 2.2.4. Forma de asignación individual estática
- 2.3. Tipos y declaraciones
 - 2.3.1. Expresiones de tipos y equivalencias
 - 2.3.2. Declaraciones y distribución de almacenamiento
 - 2.3.3. Secuencias de las declaraciones
 - 2.3.4. Campos en registros
- 2.4. Traducción de expresiones
 - 2.4.1. Operaciones dentro de expresiones
 - 2.4.2. Traducción incremental
 - 2.4.3. Direccionamiento de los elementos de un arreglo
 - 2.4.4. Traducción de referencias a arreglos
- 2.5. Comprobación de tipos
 - 2.5.1. Reglas para la comprobación de tipos
 - 2.5.2. Conversiones de tipos
 - 2.5.3. Sobrecarga de funciones y operadores
 - 2.5.4. Inferencia de tipos y funciones polimórficas
 - 2.5.5. Un algoritmo para la unificación
- 2.6. Flujo de control
 - 2.6.1. Expresiones booleanas
 - 2.6.2. Código de corto circuito
 - 2.6.3. Instrucciones de flujo de control
 - 2.6.4. Traducción del flujo de control de las expresiones booleanas
 - 2.6.5. Evitar goto redundantes
 - 2.6.6. Valores booleanos y código de salto
- 2.7. Parcheo de retroceso
 - 2.7.1. Generación de código de una pasada
 - 2.7.2. Técnica de retroceso
 - 2.7.3. Instrucciones de flujo de control
- 2.8. Instrucciones switch
 - 2.8.1. Traducciones de switch
 - 2.8.2. Traducción orientada por la sintaxis de switch
- 2.9. Código intermedio para procedimientos

UNIDAD 3: ENTORNO DE EJECUCIÓN Y OPTIMIZACIÓN DE CÓDIGO (10 PERIODOS)

Competencia específica: optimiza el código de tres direcciones para obtener un código eficiente y que esté listo para generar código objeto.

- 3. Entorno de ejecución y optimización de código
 - 3.1. Entorno de ejecución

- 3.2. Bloques básicos y grafos de flujo
 - 3.2.1. Bloques básicos
 - 3.2.2. Información de siguiente uso
 - 3.2.3. Grafos de flujo
 - 3.2.4. Representación de los grafos de flujo
 - 3.2.5. Ciclos
- 3.3. Optimización de los bloques básicos
 - 3.3.1. Representación GDA
 - 3.3.2. Búsqueda de subexpresiones locales comunes
 - 3.3.3. Eliminación de código muerto
 - 3.3.4. Uso de identidades algebraicas
 - 3.3.5. Representación de referencias a arreglos
 - 3.3.6. Asignación de apuntadores y llamadas a procedimientos
 - 3.3.7. Reensamblado de bloques básicos
- 3.4. Optimización de mirilla
 - 3.4.1. Eliminación de instrucciones redundantes
 - 3.4.2. Eliminación de código inalcanzable
 - 3.4.3. Optimizaciones de flujo de control
 - 3.4.4. Simplificación algebraica y reducción por fuerza

VIII. CONTENIDO PROCEDIMENTAL

UNIDAD 1:

- Descripción de conceptos
- Análisis de diferentes tipos de analizadores
- Traducción dirigida por la sintaxis

UNIDAD 2:

- Conocimiento del código de tres direcciones
- Generación de código de tres direcciones

UNIDAD 3:

- Conceptos de bloques básicos
- Optimización de código intermedio

IX. CONTENIDO ACTITUDINAL

Para todas las unidades: se interesa por el contenido, participa activamente y promueve el aporte del grupo.

X. ACTIVIDADES DE LABORATORIO

Proyectos de laboratorio

Proyecto Entrega 1

Duración: 14 días

Fecha aproximada de entrega: 15 de junio

Tema principal: traducción dirigida por la sintaxis tanto de analizadores sintácticos ascendentes como descendentes.

Proyecto Entrega 2

Duración: 14 días

Fecha aproximada de entrega: 29 de junio

Tema principal: generación de código intermedio, optimización y ejecución.

XI. INDICADORES DE LOGRO

- Describe conceptos de lenguajes de programación y formales
- Analiza adecuadamente los diferentes paradigmas
- Clasifica correctamente lenguajes de programación
- Identifica los tipos de gramática
- Elabora gramáticas de manera correcta
- Construye analizadores sintácticos
- Traduce un lenguaje de alto nivel a uno de bajo de nivel
- Interpreta instrucciones de alto nivel.

XII. PRODUCTOS DE APRENDIZAJE

Según el Reglamento General de Evaluación y Promoción del Estudiante de la Universidad de San Carlos de Guatemala, la zona tiene valor de 75 puntos, la nota mínima de promoción es de 61 puntos y la zona mínima para optar a examen final es de 36 puntos. Además, los laboratorios se deben aprobar con el 61% de la nota.

3 parciales	45 puntos
Tareas, prácticas, otras actividades	05 puntos
Laboratorio	25 puntos

Zona	75 puntos
Examen final	25 puntos

Nota final	100 puntos

XIII. CRITERIOS DE EVALUACIÓN

Conceptual declarativo: demuestra conceptos teóricos y prácticos en la construcción de compiladores.

Procedimental: aplica los conocimientos adquiridos de los analizadores en hojas de trabajo, ejercicios, trabajo escritos y cursos en línea.

Actitudinal: participa activamente en la clase evaluado mediante la observación.

XIV. RECURSOS DIDÁCTICOS

- Pizarrón
- Marcadores y almohadilla
- Cañonera y presentaciones multimedia
- Libros de texto
- Documentos en Internet

XV. BIBLIOGRAFÍA Y RECURSOS

Libros de texto:

- Compiladores. Principios, Técnicas y Herramientas Aho, Sethi y Ullmam. Pearson Addison-Wesley, 2008, segunda edición.

Curso tutorial de Javascript y página oficial de Jison:

- <https://www.sololearn.com/learning/1024>
- <https://zaa.ch/jison/>