DATOS DE IDENTIFICACIÓN DEL CURSO

DEPARTAMENTO:	CIENCIAS COMPUTACIONALES				
ACADEMIA A LA QUE PERTENECE:	SOFTWARE DE SISTEMAS				
NOMBRE DE LA MATERIA:	TALLER DE PROGRAMACION DE SISTEMAS				
CLAVE DE LA MATERIA:	CC207				
CARÁCTER DEL CURSO:	BÁSICA PARTICULAR				
TIPO DE CURSO:	TALLER				
No. DE CRÉDITOS:	4				
No. DE HORAS TOTALES:	60	**Presencial**	51	**No presencial**	9
ANTECEDENTES:	Se sugiere cursar de manera simultánea con CC206				
CONSECUENTES:	CC317 y CC318				
CARRERAS EN QUE SE IMPARTE:	Licenciatura en Informática Ingeniería en Computación Licenciatura en Matemáticas				
FECHA DE ULTIMA REVISIÓN:	20 de agosto de 2008				

PROPÓSITO GENERAL

Aplicar los conocimientos y habilidades como son la programación, el diseño de autómatas, el manejo de bases numéricas, el manejo de archivos de texto. Para diseñar un ensamblador utilizando la arquitectura HC12.

OBJETIVO TERMINAL

EL ALUMNO DISEÑARÁ Y PROGRAMARÁ UN ENSAMBLADOR APLICANDO LAS PRINCIPALES TÉCNICAS DE DISEÑO Y GENERACIÓN DE PROGRAMAS DE SISTEMAS POR COMPUTADORA, EMPLEANDO ALGUN LENGUAJE DE ALTO NIVEL.

CONOCIMIENTOS PREVIOS

Aplicar y emplear algún estilo de programación.
Manejar y programar en algún lenguaje de alto nivel.
Conocer la Arquitectura básica de una computadora.
Aplicar y emplear la estructura de datos (algoritmos de búsqueda y ordenación).
Aplicar y emplear archivos.
Aplicar y emplear bases numéricas.
Aplicar y emplear combinacional.
Aplicar y emplear diseño de autómatas.
Aplicar y emplear el manejo de cadenas de caracteres.
Aplicar y emplear aritmética binaria.

HABILIDADES Y DESTREZAS A DESARROLLAR
Habilidad para diseñar programas de sistemas como es el ensamblador.

ACTITUDES Y VALORES A FOMENTAR
Trabajo en equipo y trabajo colaborativo.

METODOLOGÍA DE ENSEÑANZA APRENDIZAJE								
Método	**Método tradicional de exposición**	**Método Audiovisual**	**Aula Interactiva**	**Multimedia**	**Desarrollo de proyecto**	**Dinámicas**	**Estudio de casos**	**Otros (Especificar)**
100%	**33%**				**67%**			

CONTENIDO TEMÁTICO

MODULO 1. Modelo de programación de la arquitectura de la computadora.			**3 HRS**
OBJETIVO DEL MODULO: Identificar con detalle los procesos lógicos que suceden en la arquitectura de una computadora para que pueda ejecutar un programa en lenguaje máquina.			
1.1	**Lenguaje máquina de la arquitectura de computadora.**		**1 HRS**
	OBJETIVO DEL TEMA. Identificar el lenguaje máquina de la arquitectura mediante ejercicios.		
1.2	**Modelo de programación de la arquitectura de computadora.**		**2 HRS**
	OBJETIVO DEL TEMA. Identificar el modelo de programación de la arquitectura, analizando la arquitectura de la máquina		
MODULO 2. Ensambladores.			**30 HRS**
OBJETIVO DEL MODULO. Diseñar un ensamblador de 2 pasos manejando los diferentes esquemas de ensamblado.			
2.1	**Paso Uno del Algoritmo de Dos Pasos.**		**12 HRS**
	OBJETIVO DEL TEMA. Implementar el algoritmo del 1er paso del ensamblador empleando los archivos auxiliares		
	2.1.1	**Sintaxis de las líneas de código en lenguaje ensamblador.**	**3**
		OBJETIVO DEL SUBTEMA. Implementar un programa para separar las partes de una línea en lenguaje ensamblador previo análisis de la misma.	
	2.1.2	**Instrucciones y el conjunto de instrucciones.**	**3**
		OBJETIVO DEL SUBTEMA. Modificar el programa que se diseñó en el punto 2.1.1. El programa actualizado, además, deberá de realizar la búsqueda de cada instrucción identificada por cada línea y determinar cuales modos de direccionamiento le corresponde a cada una de ellas de acuerdo a la tabla de códigos de operación.	
	2.1.3	**Instrucciones y modo de direccionamiento.**	**3**
		OBJETIVO DEL SUBTEMA. Modificar el programa que se diseño en el punto 2.1.2. El programa actualizado además, deberá de obtener el modo de direccionamiento particular de acuerdo al operando, considerando las	

		diversas bases numéricas y tamaño de la instrucción en bytes.	
	2.1.4	**Directivas, contador de localidades y archivos auxiliares.**	**3**
		OBJETIVO DEL SUBTEMA. Modificar el programa que se diseño en el punto 2.1.3. El programa actualizado además, deberá de validar la sintaxis de las directivas; generar el contador de localidades; y generar los archivos auxiliares (temporal o de listado; y tabla de símbolos).	
2.2	**Paso Dos del Algoritmo de Dos Pasos.**		**18 HRS**
	OBJETIVO DEL TEMA. Implementar el algoritmo del 2do paso del ensamblador empleando los archivos auxiliares.		
	2.2.1	**Modos de direccionamiento simples.**	**3**
		OBJETIVO DEL SUBTEMA. Modificar el programa que se diseño en el punto 2.1.4. El programa actualizado además, deberá generar el código máquina de los direccionamientos simples.	
	2.2.2	**Modos de direccionamiento indizados y relativos.**	**9**
		OBJETIVO DEL SUBTEMA. Obtener el código máquina de direccionamientos indizados y los relativos	
	2.2.2.1	**Modos de direccionamiento indizados simples.**	**3**
		OBJETIVO DEL SUBTEMA. Modificar el programa que se diseño en el punto 2.2.2. El programa actualizado además, deberá generar el código máquina de los direccionamientos indexados simples (5, 9,16 bits).	
	2.2.2.2	**Modos de direccionamiento indizados complejos.**	**3**
		OBJETIVO DEL SUBTEMA. Modificar el programa que se diseño en el punto 2.2.2.1. El programa actualizado además, deberá generar el código máquina de los direccionamientos indexados complejos.	
	2.2.2.3	**Modos de direccionamiento indizados relativos.**	**3**
		OBJETIVO DEL SUBTEMA. Modificar el programa que se diseño en el punto 2.2.2.2. El programa actualizado además, deberá generar el código máquina de los direccionamientos indexados relativos.	
	2.2.3	**Instrucciones que utilizan etiquetas y directivas del lenguaje ensamblador.**	**3**
		OBJETIVO DEL SUBTEMA. Modificar el programa que se diseño en el punto 2.2.2.3. El programa actualizado además, deberá generar el código máquina de instrucciones que utilizan etiquetas y de las directivas del lenguaje ensamblador.	
	2.2.4	**Código objeto.**	**3**
		OBJETIVO DEL SUBTEMA. Modificar el programa que se diseño en el punto 2.2.3. El programa actualizado además, deberá generar archivo objeto.	

2.3	**Ensamblador.**	**12 HRS**
	OBJETIVO DEL TEMA. Demostrar el funcionamiento del ensamblador diseñado previamente.	

MODULO 3. Cargador y Ligador.		**9 HRS**
OBJETIVO DEL MODULO. Diseñar el pseudocódigo de un cargador; y evaluar los algoritmos de ligadores existentes		
2.1	**Cargador.**	**6 HRS**
	OBJETIVO DEL TEMA. Diseñar el pseudocódigo del cargador absoluto.	
2.2	**Ligador**	**3 HRS**
	OBJETIVO DEL TEMA. Identificar las ventajas y las desventajas de los diferentes algoritmos de ligado.	

CRITERIOS DE EVALUACIÓN	
PROYECTO FINAL	40%
PRACTICAS-TAREAS (12)	60%

- 10 prácticas del ensamblador: 5 puntos por práctica, con su reporte (3 práctica, 2 reporte). En total son 50 puntos.
- Ensamblador completo: 40 puntos. El ensamblador deberá incluir todos los modos de direccionamiento y directivas. En caso contrario por cada exclusión será 5puntos menos de puntaje. (Esto es por modo de direccionamiento: cada tipo de indexado y relativo cuentan como 1 modo independiente).
- 2 prácticas del cargador y ligador: 5 puntos cada uno. En total son 10 puntos.

NOTA:

- Las prácticas-tareas son para ENTREGARSE personalmente el día marcado en hojas tamaño carta con los siguientes datos: nombre de la materia, nombre del alumno, fecha de entrega, numero de tarea y desarrollo de la misma. NO-SE RECIBIRAN PRÁCTICAS ATRASADAS.
- El ensamblador puede ser entregado en equipo (máximo 2 personas) al ser entregado deberá incluir un reporte de la explicación del mismo (modelo de programación, ligas de objetos, tabla de símbolos en su caso, uso de funciones C++, etc) con los datos arriba mencionados.
- Para poder evaluar el proyecto final, deberá contar con el 80% de asistencia y tener actividades registradas durante el curso.
-

BIBLIOGRAFÍA

BÁSICA

TITULO	AUTOR	EDITORIAL	AÑO DE EDICIÓN	% DE COBERTURA DEL CURSO
System Software. An Introduction to systems programming	Leland L. Beck	Addison Wesley	1997	100
Software de sistemas Introducción a la programación de Sistemas.	Leland L. Beck	Addison Wesley	1988	100

COMPLEMENTARIA

TITULO	AUTOR	EDITORIAL	AÑO DE EDICIÓN	% DE COBERTURA DEL CURSO
Lenguaje ensamblador y programación para PC IBM y compatibles	Peter Abel	Prentice Hall	1996	33
Los microprocesadores Intel Arquitectura, programación e interfaz de los procesadores 8086/8088/80186/80188, 80286, 80386, 80486, Pentium, Pentium Pro y Pentium II	Barry B. Brey	Pearson Educación	2001	33
Lenguaje ensamblador para microcomputadoras IBM. Para principiantes y avanzados.	J. Terry Godfrey	Prentice Hall Hispanoamericana, S.A	1991	33

REVISIÓN REALIZADA POR:

NOMBRE DEL PROFESOR	FIRMA
María Elena Romero Gastelú	
Norma Ramírez Hernández	
Valentín Martínez López	
Francisco Javier Quintanilla Moreno	
Víctor Manuel Zamora Ramos	
Ramiro Alonso Vera Contreras	
Luis Angel Morales López	

UNIVERSIDAD DE GUADALAJARA

CENTRO UNIVERSITARIO DE CIENCIAS EXACTAS E INGENIERÍAS
DIVISIÓN DE ELECTRÓNICA Y COMPUTACIÓN

Vo.Bo. Presidente de Academia	Vo.Bo. Jefe del Departamento

lunes, 08 de septiembre de 2008

Av. Revolución No. 1500, Módulo O, Planta Baja, S.R. C.P. 44840, Guadalajara, Jal., México. Tel/Fax (33) 1378 5900 ext. 7732
http://dcc.cucei.udg.mx/