



Programa de estudio

Datos generales

Área Académica

Económica Administrativa

Programa educativo

Licenciatura en Informática

Facultad

Estadística e Informática

Código

17

Nombre de la experiencia educativa

Arquitectura de Computadoras II

Área curricular

Básica general	Iniciación a la disciplina X	Disciplinar	Terminal	Electiva
----------------	------------------------------	-------------	----------	----------

Proyecto integrador.

Desarrollo y aplicación de tecnología computacional y de telecomunicaciones

Academia(s)

Arquitectura de Computadoras

Requisito (s)

Prerrequisito(s): Arquitectura de Computadoras I

Correquisito (s):

Modalidad

Curso

Características del proceso de enseñanza aprendizaje

Individual	Grupal X	Número mínimo:10
		Número máximo: 50

Número de horas de la experiencia educativa

Teóricas: 4

Prácticas: 2

Total de créditos

10

Total de horas

90

Equivalencias

Fecha de elaboración/modificación

a. 29 /julio/2002

Fecha de aprobación

17/noviembre/2002

b. 19 /Septiembre/2003

Nombre de los académicos que participaron en la elaboración y/o modificación.

José Rafael Rojano Cáceres, V. Angélica García Vega, Sergio Castillo Valerio, José de Jesús Martínez

Perfil del docente

Ingeniero o Licenciado en Informática o Computación o carrera afín, con maestría en el área, preferentemente

Espacio

Relación disciplinar

Intra Programa Educativo.

Interdisciplinario.

Descripción mínima

Conocimiento y comprensión de los procesos de intercambio de información a bajo nivel por medio de instrucciones en lenguaje máquina en la arquitectura de los procesadores x86 de la familia Intel y compatibles, de manera que pueda programarlos para fines específicos, mostrando actitudes de colaboración, meticulosidad y creatividad.

Justificación

Las ciencias de la computación exigen por parte de quienes se dedican a ella entre diversas habilidades, la de programar. Esta habilidad debe ser sustentada no solo en un lenguaje de alto nivel específico, sino también en un profundo conocimiento de la relación hardware y software, proporcionado a través del lenguaje ensamblador.

La importancia de esta experiencia no solo se justifica en el aprendizaje de un lenguaje ensamblador como herramienta de comprensión del nivel hardware, sino también de experimentación científica, y de soporte para la optimización en la ingeniería de software.

Unidad de Competencia

En un ambiente de colaboración y con creatividad, honestidad, responsabilidad y autoaprendizaje el alumno:

- Conoce y Comprende: Los procesos de intercambio de información de bajo nivel mediante instrucciones de máquina, las diferencias que existen entre los diversos tipos de memoria y la aplicación de estas, la organización de la memoria de una computadora y como se lleva a cabo la lectura y escritura de información para uso de la UCP, la estructura interna de un procesador, como se llevan a cabo, por medio de circuitos lógicos, las instrucciones de máquina en un procesador, las formas de comunicación e identificación de la UCP con los dispositivos periféricos, la diferencia que existe entre las diversas arquitecturas de procesadores existentes en el mercado.
- Ensambla y desarrolla programas en lenguaje ensamblador.
- Diseña con compuertas lógicas Unidades Aritméticas Lógicas.
- Identifica: Instrucciones escritas en lenguaje máquina y hace traducciones de lenguaje ensamblador a máquina y viceversa, y los diferentes procedimientos y dispositivos para la atención de los periféricos por la UCP.

Articulación con los ejes

Teóricos	Heurísticos	Axiológicos
<p>1. Concepto básicos</p> <p>1.1. Definir que es el lenguaje máquina</p> <p>1.1.1. Código de operación</p> <p>1.1.2. Conjunto de instrucciones</p> <p>1.1.3. Comparación Arquitecturas RISC y CISC</p> <p>1.2. Definir que es el lenguaje ensamblador</p> <p>1.2.1. Mnemónico</p> <p>1.3. Características de ambos lenguajes</p> <p>1.4. Diferencia entre dato e instrucción</p> <p>1.5. Formato de instrucción en el ensamblador</p> <p>1.6. Repaso de bit, byte, nibble, etc.</p>	<ul style="list-style-type: none"> •Comprende la diferencia entre el lenguaje de máquina y el lenguaje ensamblador. •Conoce el formato de las instrucciones del lenguaje ensamblador. •Comprende la diferencia entre bits y bytes, así como otras formas de agrupación. •Distingue y reconoce las diferencias entre los procesadores de cada arquitectura RISC y CISC. 	<ul style="list-style-type: none"> •Colaboración •Creatividad •Autoaprendizaje •Discreción •Responsabilidad •Honestidad
<p>2. Introducción a la arquitectura de un procesador genérico (familia x86)</p> <p>2.1. Componentes del procesador</p> <p>2.1.1. Registros de la Cpu</p> <p>2.1.1.1. Propósito general</p> <p>2.1.1.2. Especiales: Segmento, índices, apuntadores de pila, apuntador de instrucción</p> <p>2.1.1.3. Registro de códigos de condición: Bandera</p> <p>2.1.2. Canal de datos, de dirección, de control</p> <p>2.2. Ciclo de instrucción: Fetch/Execute</p> <p>2.3. Modelos de gestión de memoria del x86</p> <p>2.3.1. Real</p> <p>2.3.2. Segmentado</p> <p>2.3.3. Plano</p> <p>2.4. Modos de operación del procesador (Real, protegido y de administración).</p> <p>2.5. Modos de direccionamiento</p> <p>2.5.1. Modos directo, indirecto, inmediato de direccionamiento de la arquitectura x86 Intel</p> <p>2.5.2. Diferencia entre valor y localidad de memoria</p>	<ul style="list-style-type: none"> •Conoce un modelo abstracto de los registros de la CPU y los modos de direccionamiento. •Reconoce las etapas del ciclo de instrucción. •Asocia los elementos del procesador utilizados en cada una de las etapas del ciclo de instrucción. •Conoce los modelos de memoria que maneja una arquitectura como la x86. •Reconoce los componentes abstractos más comunes y específicos de los procesadores. 	
<p>3. Introducción al ensamblador</p> <p>3.1. Herramientas de programación en lenguaje máquina</p> <p>3.1.1. Captura de programas</p> <p>3.1.2. Ejecución paso a paso de programas</p> <p>3.1.3. Ejecución continua</p> <p>3.2. Panorámica del conjunto de instrucciones: aritméticas, lógicas, transferencia, corrimientos y rotaciones.</p> <p>3.3. Representaciones y codificaciones de datos numéricos y caracteres:</p> <p>3.3.1. Representación posicional, complemento a 1, complemento a 2, punto flotante.</p> <p>3.3.2. Códigos ASCII, EBCDIC, etc.</p> <p>3.3.3. Subconjunto de instrucciones y directivas más usadas (Mov, Add, Jmp, etc.)</p> <p>3.4. Introducción y uso de las interrupciones</p> <p>3.4.1. Interrupciones</p>	<ul style="list-style-type: none"> •Es capaz de manejar el entorno de programación de bajo nivel. •Aplica los conceptos aprendidos en las unidades 1 y 2, teniendo interacción con la arquitectura de la computadora y el ensamblador a través de una herramienta de programación de bajo nivel. •Conoce el subconjunto de instrucciones y directivas más comunes. •Es capaz de aplicar en la práctica los modos de direccionamiento explicados en la unidad 2. •Distingue las diferencias entre las interrupciones de software y de hardware. •Es capaz de asociar y describir el proceso que se lleva a cabo en el entorno de bajo nivel. 	

<p>3.4.2. Interrupciones vía Software vs interrupciones vía Hardware.</p> <p>3.4.3. Relación y uso de los registros y las interrupciones</p> <p>3.4.4. Uso de interrupciones del BIOS y del DOS para captura y despliegue de caracteres</p>	
<p>4. Introducción a la sintaxis de un Macro ensamblador</p> <p>4.1. Diferencias entre los programas del tipo .com y los .exe</p> <p>4.2. Directivas simplificadas y extendidas</p> <p>4.2.1. Manejo de memoria.</p> <p>4.2.2. Definición de segmentos.</p> <p>4.2.3. Definición de variables y constantes</p> <p>4.2.4. Definición de procedimientos.</p>	<ul style="list-style-type: none"> •Conoce la sintaxis usada para escribir programas en un Macro ensamblador. •Diseña y programa en lenguaje ensamblador pequeños programas para resolver problemas sencillos. •Comprende y describe las fases de la generación de programas ejecutables. •Comprende la diferencia entre código absoluto y código relocizable. •Comprende las diferencias y semejanzas entre los distintos programas ejecutables.
<p>5. Macro y procedimientos</p> <p>5.1. Macros</p> <p>5.1.1. Instrucciones y directivas para la escritura de macros.</p> <p>5.2. Procedimientos</p> <p>5.2.1. Instrucciones y directivas para la escritura de procedimientos.</p> <p>5.3. Semejanzas y diferencias entre macros y procedimientos.</p> <p>5.4. Paso de parámetros</p> <p>5.4.1. Pila</p> <p>5.4.2. Registros</p>	<ul style="list-style-type: none"> •Comprende la diferencia entre escribir macros y procedimientos. •Conoce como el ensamblador resuelve por pasadas la llamadas a macroinstrucciones y a procedimientos. •Entiende la importancia de la estructura pila en la computación (para la recursión) y para el paso de parámetros •Asocia los conceptos de paso de parámetros por valor y por referencia con el uso de pila y registro.
<p>6. Instrucciones complejas de ensamblador</p> <p>6.1. Manejo de cadenas en ensamblador</p> <p>6.2. Control de ciclos</p>	<ul style="list-style-type: none"> •Es capaz de usar y manipular instrucciones complejas y su relación con las banderas.
<p>7. Manipulación de dispositivos de entrada y salida</p> <p>7.1. Manipulación por interrupciones de Bios y Dos</p> <p>7.2. Manipulación por control directo</p> <p>7.2.1. Dispositivos mapeados a memoria</p> <p>7.2.2. Dispositivos con puertos de entrada y salida</p>	<ul style="list-style-type: none"> •Describe y evalúa la complejidad entre usar las interrupciones del BIOS, del DOS y el manejo directo de los dispositivos. •Asocia el manejo de los dispositivos en bajo nivel y en alto nivel a través del uso del sistema operativo.
<p>8. Manejo de un dispositivo con puertos de entrada y salida</p> <p>8.1. Especificación del puerto paralelo estándar (spp)</p> <p>8.2. Construcción de un cable paralelo</p> <p>8.3. Instrucciones para escritura y lectura del puerto</p> <p>8.3.1. Debug</p> <p>8.3.1.1. Entrada (i)</p> <p>8.3.1.2. Salida (o)</p> <p>8.3.2. Macro-ensamblador</p> <p>8.3.2.1. In</p> <p>8.3.2.2. Out</p> <p>8.4. Direcciones de acceso al puerto paralelo estándar</p> <p>8.5. Bytes de acceso al puerto paralelo estándar</p> <p>8.5.1. Byte de datos (salida)</p>	<ul style="list-style-type: none"> •Comprende y aplica la forma de interactuar con dispositivos periféricos a través de los puertos. •Emplea las instrucciones de lenguaje ensamblador, así como los comandos de bajo nivel para escribir y leer a través del puerto paralelo. •Conoce las diferentes especificaciones para el puerto paralelo EPP, ECP, etc. •Conoce los rangos de entrada salida para el puerto paralelo estándar bajo el D.O.S.

<p>8.5.2. Byte de estado (entrada)</p> <p>8.5.3. Byte de control (salida)</p> <p>8.6. Realizar prácticas con la protoboard y el SPP de la PC.</p> <p>8.6.1. Entrada de datos</p> <p>8.6.2. Salida de datos</p>	
<p>9. Manejo de un dispositivo mapeado a memoria</p> <p>9.1. Uso de la memoria de video</p> <p>9.2. Acceso directo a la memoria de video</p> <p>9.3. Rangos de entrada y salida para esta memoria</p> <p>9.4. Acceso al modo gráfico</p> <p>9.5. Modos gráficos</p> <p>9.5.1. Int 10</p> <p>9.5.2. Desplazamiento del cursor</p> <p>9.5.3. Dibujado de formas con pixeles</p>	<ul style="list-style-type: none"> •Es capaz de acceder y manipular al modo gráfico para el despliegue de datos, a través de la memoria de video. •Es capaz de desarrollar pequeños programas en el modo gráfico.
<p>10. Tópicos especiales.</p> <p>10.1. Programación desde lenguaje de alto nivel y ensamblador.</p> <p>10.1.1. programación en línea.</p> <p>10.1.2. módulos en ensamblador.</p> <p>10.1.2.1. paso de parámetros.</p> <p>10.2. Introducción a la programación de 32 bits.</p> <p>10.2.1. Directivas</p> <p>10.2.2. Modelo de memoria</p> <p>10.2.3. API de Windows</p> <p>10.3. Registros relacionados a la programación de 32 bits.</p> <p>10.4. Ventajas de la programación de 32 bits.</p>	<ul style="list-style-type: none"> •Comprende las limitaciones de la programación en ambientes de 16 bits versus los de 32 bits. •Desarrolla alguna aplicación pequeña para conocer de la programación de 32 bits (despliegue de ventanas, captura de texto, etc).

Materiales didácticos	Recursos didácticos
1) 8088-8086/8087 : programación ensamblador en entorno MS DOS. <i>Rodríguez-Roselló, Miguel Angel.</i> Madrid : Anaya Multimedia, 1993 2) Lenguaje ensamblador de los 80x86. <i>Beltrán de Heredia, Jon.</i> Madrid : Anaya, 1998 3) 80386/80286 programación en lenguaje ensamblador. <i>Murray , William H.,</i> México : McGraw-Hill, 1987 4) Lenguaje ensamblador para microcomputadoras IBM : para principiantes y avanzados. <i>Godfrey, J. Terry.</i> México : Prentice-Hall Hispanoamericana, 1991 5) INTRODUCCIÓN AL PROCESADOR 8088/8086. <i>Cristopher L. Morgan y Mitchel Waite</i> Ed. McGraw Hill. Primera edición en español, México 1988 6) Organización de computadoras, un enfoque estructurado. <i>Andrew Tanenbaun.</i> México Prentice Hall, segunda edición, México, 1985 7) Organización de computadoras. <i>Hamacher, Vranesic y Zaky.</i> México : McGraw-Hill, c1988 8) IBM PC Assembly Lenguaje. <i>Dona S. Tabler John Wily & Sons, Inc,</i> 1985 9) Programación de Microcomputadoras Basada en el procesador 8086/8088. <i>Ramón Cortés Barrios.</i> Limusa, 1989 10) Introducción a un Lenguaje Ensamblador para Arquitectura de Computadoras. <i>Karen Millar.</i> Oxford, 1999. 11) Manuales de prácticas. 12) Manuales técnicos de componentes	1. Computadoras. 2. Cañón. 3. Componentes TTL : compuertas lógicas y Flip-Flops. 4. Fuentes de poder. 5. Multímetros. 6. Programas de cómputo (compiladores MASM y NASM). 7. Laboratorios. 8. Tarjetas de experimentación (Protoboard). 9. Pinzas. 10. Cable UTP. 11. Cautines. 12. Conectores y conchas (DB 25). 13. Leds. 14. Resistencias. 15. Condesadotes. 16. Osciloscopios. 17. Otros materiales

Evaluación del desempeño

Evidencia(s) de desempeño	Criterios de desempeño	Campo(s) de aplicación	Porcentaje
<ul style="list-style-type: none"> • Bitácoras • Reportes • Montaje de circuitos • Programas corriendo eficientemente en lenguaje ensamblador. • Puntualidad 	<ul style="list-style-type: none"> • Entrega puntual de las reseñas críticas • Elaboración y entrega de los reportes de las prácticas • Puntualidad e intervenciones en clases • Aprobación de los exámenes parciales • La calificación se distribuirá de la siguiente manera: • 60 % trabajo en clase, reportes y prácticas 40% calificación aprobatoria de los exámenes 	Aula Taller	70 30

Acreditación

Para acreditar esta experiencia el alumno deberá demostrar sus conocimientos teóricos y prácticos del

tema:

- Presentando todas las prácticas sugeridas por el profesor en tiempo y forma.
- Aprobando los exámenes teóricos/prácticos a lo largo del periodo, y
- Cumpliendo con el mínimo de asistencias estipulados en el reglamento.

Fuentes de información Básicas

- 1) **8088-8086/8087 : programación ensamblador en entorno MS DOS.** *Rodríguez-Roselló, Miguel Angel.* Madrid : Anaya Multimedia, 1993
- 2) **Lenguaje ensamblador de los 80x86.** *Beltrán de Heredia, Jon.* Madrid : Anaya, 1998
- 3) **80386/80286 programación en lenguaje ensamblador.** *Murray , William H.,* México : McGraw-Hill, 1987
- 4) **Lenguaje ensamblador para microcomputadoras IBM : para principiantes y avanzados.** *Godfrey, J. Terry.* México : Prentice-Hall Hispanoamericana, 1991
- 5) **INTRODUCCIÓN AL PROCESADOR 8088/8086.** *Cristopher L. Morgan y Mitchel Waite* Ed. McGraw Hill. Primera edición en español, México 1988
- 6) **Organización de computadoras, un enfoque estructurado.** *Andrew Tanenbaun.* México Prentice Hall, segunda edición, México, 1985

Complementarias

Sitios de Internet, reportes específicos de los productores de equipos tales como:

<http://www.ctyme.com/rbrown.htm> (Listado de todas las interrupciones para ensamblador, muy completo)

<http://www.intel.com> (Contiene documentos esenciales para la materia en cuanto a las especificaciones técnicas del hardware)

<http://www.beyondlogic.com>

<http://developer.intel.com/design/pentiumii/manuals/243190.htm>