

1.- DATOS DE LA ASIGNATURA

| |
|--|
| Nombre de la asignatura: Lenguaje ensamblador |
| Carrera: Ingeniería en Sistemas Computacionales |
| Clave de la asignatura: SCM - 0420 |
| Horas teoría-horas práctica-créditos 3-2-8 |

2.- HISTORIA DEL PROGRAMA

| Lugar y fecha de elaboración o revisión | Participantes | Observaciones (cambios y justificación) |
|--|---|---|
| Instituto Tecnológico de Toluca del 18 al 22 agosto 2003. | Representantes de la academia de sistemas y computación de los Institutos Tecnológicos. | Reunión nacional de evaluación curricular de la carrera de Ingeniería en Sistemas Computacionales. |
| Instituto Tecnológico de: Cancún, Cd. Victoria, Nuevo León, Zitácuaro. 23 agosto al 7 de noviembre 2003. | Academia de sistemas y computación. | Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación. |
| Instituto Tecnológico de León 1 al 5 de marzo 2004. | Comité de consolidación de la carrera de Ingeniería en Sistemas Computacionales. | Definición de los programas de estudio de la carrera de Ingeniería en Sistemas Computacionales. |

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

| Anteriores | | Posteriores | |
|-------------------------------|---|-------------|---|
| Asignaturas | Temas | Asignaturas | Temas |
| Arquitectura de computadoras. | Comunicación interna de la computadora. | Interfaces. | Periféricos estandarizados Interfaces. -Programación de bajo nivel. |

b). Aportación de la asignatura al perfil del egresado

Conoce el control de tareas en hardware y software por medio de un lenguaje de bajo nivel.

4.- OBJETIVO(S) GENERAL(ES) DEL CURSO

El estudiante utilizará un lenguaje de bajo nivel para saber explotar las capacidades de hardware que dispone un equipo de cómputo, operando eficientemente sus componentes

5.- TEMARIO

| Unidad | Temas | Subtemas |
|--------|------------------------|---|
| 1 | Fundamentos. | 1.1 Introducción. 1.1.1 Uso y aplicaciones del lenguaje ensamblador. 1.1.2 Escalabilidad de los microprocesadores. 1.1.3 Tipos de lenguajes ensambladores. 1.1.4 Clasificación de Memorias. 1.1.5 Unidades de entrada / salida. 1.2 El microprocesador. 1.2.1 Buses. 1.2.2 Registros. 1.2.3 Modos de direccionamiento. 1.3 Interrupciones. 1.3.1 Hardware. 1.3.2 Software. 1.4 Estructura de un programa en ensamblador. 1.4.1 Data segment. 1.4.2 <u>S</u> nack segment. 1.4.3 Code segment. 1.4.4 Instrucciones del programa. 1.4.5 Directivas. 1.5 Procedimiento de ensamble, enlace y ejecución. 1.6 Entorno de programación. |
| 2 | Elementos del lenguaje | 2.1 Instrucciones lineales. 2.1.1 Movimiento. 2.1.2 Pila. 2.1.3 Matemáticos. 2.1.4 Ajustes. 2.1.5 Comparación. 2.2 Saltos. 2.2.1 Incondicional. 2.2.2 Condicional. 2.3 Tipos de ciclos. 2.4 Operadores Lógicos. 2.5 Desplazamiento. 2.5.1 Lineal. 2.5.2 Circular. |

5.- TEMARIO (Continuación)

| | | |
|---|-----------------------|--|
| 3 | Modularización | 2.6 Procesos de control. 2.6.1 Banderas. 2.6.2 Cadenas. 2.6.3 Carga. |
| 4 | Programación híbrida. | 3.1 Macros. 3.1.1 Internas. 3.1.2 Externas. 3.2 Procedimientos. 3.2.1 Internos. 3.2.2 Externos. 4.1 Directivas para compilación híbrida. 4.2 Funciones en ensamblador. 4.3 Bloques en ensamblador. 4.4 Operadores. 4.5 Integrar módulos de ensamblador en lenguajes de alto nivel. |

6.- APRENDIZAJES REQUERIDOS

- Conocer la arquitectura de una computadora.
- poseer la capacidad de abstracción, para realizar el planteamiento y su posterior implementación haciendo uso de éste lenguaje.
- Poseer habilidades de Análisis de problemas, e identificación de componentes de hardware asociadas al software.
- Utilizar algún lenguaje de programación de alto nivel.

7.- SUGERENCIAS DIDÁCTICAS

- Revisar en dinámicas grupales los elementos del lenguaje mediante ejercicios.
- Propiciar la búsqueda y selección de información sobre programas que apliquen los elementos del lenguaje ensamblador.
- Coordinar problemas en clase y propiciar la solución en conjunto mediante un programa.
- Solicitar al alumno propuestas a problemas a resolver.

8.- SUGERENCIAS DE EVALUACIÓN

- Dar seguimiento al desempeño en el desarrollo del programa (dominio de los conceptos, capacidad de la aplicación de los conocimientos en problemas reales, transferencia del conocimiento).
- Participación en actividades individuales y de equipo.
- Cumplimiento de los objetivos y desempeño en las prácticas propuestas.
- Observar el desempeño en el desarrollo de programas.

9.- UNIDADES DE APRENDIZAJE

UNIDAD 1.- Fundamentos.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|--|---|-------------------------------|
| Conocerá las características básicas del lenguaje ensamblador y diseño interno de la computadora | <ul style="list-style-type: none">• Utilizar medios didácticos para efectuar un comparativo entre las diferentes arquitecturas de los procesadores.<ul style="list-style-type: none">a).- Unidades.b).- Registros.c).- Buses.• Buscar y comentar en clase, las marcas y modelos de procesadores para computadoras.• Analizar en clase, las diferentes interrupciones que permiten manipular el Hardware y software.• Realizar esquemas para representar el funcionamiento de las interrupciones. | 1, 7 |

UNIDAD 2.- Elementos del lenguaje.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|--|--|-------------------------------|
| Conocerá y aplicará la sintaxis del lenguaje ensamblador para manipular el equipo de computo | <ul style="list-style-type: none">• Buscar, seleccionar y comentar en clase, las características del lenguaje ensamblador.• Explicar las variantes que se pueden presentar en el uso de cada nemotécnico.• Crear un programa donde se use cada uno de ellos, con la finalidad de depurarlo explicarlo en el laboratorio. | 2, 5, 7 |

UNIDAD 3.- Modularización.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|---|--|------------------------|
| Conocerá las diferentes formas de programación modular que permite éste lenguaje. | <ul style="list-style-type: none">• Buscar, seleccionar y comentar en clase, las características del lenguaje ensamblador en el área de programación modular.• Realizar prácticas utilizando módulos en lenguaje ensamblador.• Revisar relaciones entre módulos, para evaluación de desempeño de aplicaciones. | 1, 2, 5, 6, 7 |

UNIDAD 4.- Programación híbrida.

| Objetivo Educativo | Actividades de Aprendizaje | Fuentes de Información |
|---|---|------------------------|
| Realizará módulos de lenguaje ensamblador insertándolos en programas en escritos en lenguajes de alto nivel para realizar aplicaciones. | <ul style="list-style-type: none">• Buscar, seleccionar y comentar en clase, las características de los lenguajes de alto nivel que permitan módulos en lenguajes de bajo nivel.• Realizar prácticas utilizando módulos en lenguaje ensamblador. | 1, 2, 5, 6, 7 |

10. FUENTES DE INFORMACIÓN

1. Christopher L. Morgan & Mitchell Waite. *Introducción al Microprocesador 8086/8088 (16 bits)*. Mc Graw Hill.
2. William H. Murray III & Chris H Pappas. *80386/80286 Programación en Lenguaje Ensamblador*. Mc Graw Hill.
3. Francisco Javier Cevallos. *Manual para Quick C2 Guía del Programador*. Macrobit – RaMa.
4. Ma. Isabel García Clemente, Rafael Méndez Cavanillas, Ma. Luisa Córdoba Cabeza. *Estructura de Computadores, Problemas y soluciones* AlfaOmega – RaMa.
5. J. Terry Godfrey. *Lenguaje Ensamblador para Microcomputadoras IBM para principiantes y avanzados*. Prentice Hall.
6. Borry B. Brey. *Los Microprocesadores de Intel*. Prentice Hall
7. Peter Abel. *Lenguaje Ensamblador y Programación para Pc IBM y Compatibles*. Pearson Educación.

8. A. Abiteboul, P. Buneman, And D. Suciu *Data on the Web*, Morgan - Kaufman, 1999
9. J. Han And M. Kamber *Data Mining - Concepts and Techniques* Morgan - Kaufman, 2001

11. PRÁCTICAS

Unidad Práctica

- 1 Para cada unidad adaptar las practicas propuestas en el texto de Estructura de computadores, problemas y soluciones.
- 2 Creación de programas que utilizando instrucciones lineales, ciclos y instrucciones de control.
- 3 Creación de un programa con la extensión .COM.
- 4 Crear programas que permitan manipular números y cadenas.
- 5 Crear programas que permitan crear módulos en un lenguaje de alto nivel aplicaciones en el sistema operativo.
- 6 Crear programas que permitan manipular los siguientes dispositivos: monitor, teclado, impresora, ratón, disco.