

1.- DATOS DE LA ASIGNATURA

Nombre de la asignatura: Software de sistemas
Carrera: Licenciatura en Informática
Clave de la asignatura: IFE - 0438
Horas teoría-horas práctica-créditos 2-2-6

2.- HISTORIA DEL PROGRAMA

Lugar y fecha de elaboración o revisión	Participantes	Observaciones (cambios y justificación)
Instituto Tecnológico de Puebla del 8 al 12 septiembre 2003.	Representantes de la academia de sistemas y computación de los Institutos Tecnológicos.	Reunión nacional de evaluación curricular de la carrera de Licenciatura en Informática.
Instituto Tecnológico de: Chilpancingo, León 13 septiembre al 28 de noviembre 2003.	Academia de de sistemas y computación.	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación.
Academia de de sistemas y computación.	Análisis y enriquecimiento de las propuestas de los programas diseñados en la reunión nacional de evaluación.	Academia de de sistemas y computación.

3.- UBICACIÓN DE LA ASIGNATURA

a). Relación con otras asignaturas del plan de estudio

Anteriores		Posteriores	
Asignaturas	Temas	Asignaturas	Temas
Organización de Computadoras	Funcionamiento interno de la computadora. Unidad central de proceso. Operación de la unidad de aritmética y lógica. Memorias. Interfases de entrada/salida	Sistemas Operativos I	Introducción a los sistemas operativos. Administración de procesos. Administración del procesador. Administración de la memoria. Administración de dispositivos de E/S.
Introducción a las ciencias de la computación.	Software de base.		
Estructura de datos			

b). Aportación de la asignatura al perfil del egresado

- Conoce las características del software de base.
- Desarrolla habilidades y actitudes que le permiten identificar problemas y oportunidades donde se aplique el tratamiento de información para proponer soluciones por medio de modelos y facilitar con ello la toma de decisiones.
- Selecciona y utiliza de manera óptima las herramientas computacionales actuales y emergentes.
- Realiza actividades de auditoria y consultoría relacionadas con la función informática.
- Fomenta y coordina actividades que fortalezcan la cultura informática.
- Realiza estudios de factibilidad para la selección de productos de software y hardware.

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

El estudiante comprenderá el funcionamiento del software de base para su eficiente utilización.

5.- TEMARIO

Unidad	Temas	Subtemas
1	Introducción.	<ul style="list-style-type: none">1.1 Revisión del modelo Von Newman.1.2 Del problema al programa cargado.1.3 Lenguajes formales.1.4 Editores.<ul style="list-style-type: none">1.4.1 De caracter.1.4.2 De línea.1.4.3 De pantalla.1.5 Lenguajes de programación.<ul style="list-style-type: none">1.5.1 Niveles (bajo, intermedio y alto).1.5.2 Tipos (interpretes y compiladores).1.5.3 Comparación entre intérpretes y compiladores.
2	Ensamblador.	<ul style="list-style-type: none">2.1 Importancia del lenguaje ensamblador.2.2 Manejo de memoria.2.3 Direccionamiento.2.4 Formato de un programa.2.5 Proceso de ensamble y ligado.2.6 Instrucciones<ul style="list-style-type: none">2.6.1 Instrucciones aritméticas.2.6.2 Instrucciones de comparación.2.6.3 Instrucciones de saltos.2.6.4 Instrucciones para el stack.2.7 Macros.2.8 Interrupciones.

5.- TEMARIO (Continuación)

3	Compiladores.	<p>3.1 Funciones.</p> <p>3.2 Fases de un compilador.</p> <p>3.2.1 Fase de análisis.</p> <p>3.2.1.1 Análisis lexicográfico.</p> <p>3.2.1.2 Análisis sintáctico.</p> <p>3.2.1.3 Análisis semántico.</p> <p>3.2.2 Fase de síntesis.</p> <p>3.2.2.1 Generación y optimización de código intermedio.</p> <p>3.2.2.2 Generación y optimización de código objeto.</p> <p>3.3 Diferencias entre intérpretes y compiladores.</p>
4	Ligadores y cargadores.	<p>4.1 Ligadores.</p> <p>4.1.1 Liga de bibliotecas de código objeto.</p> <p>4.1.2 Ligadores estáticos.</p> <p>4.1.3 Ligadores dinámicos.</p> <p>4.2 Cargadores.</p> <p>4.2.1 Carga absoluta y relocalizable.</p> <p>4.2.2 Cargadores estáticos.</p> <p>4.2.3 Cargadores dinámicos.</p> <p>4.3 Paso del control al sistema operativo.</p>

6.- APRENDIZAJES REQUERIDOS

- Habilidad para programar.
- Conocimiento del funcionamiento de los componentes de una computadora.
- Manejo eficiente de estructuras de datos como colas, pilas, listas y árboles.

7.- SUGERENCIAS DIDÁCTICAS

- Propiciar el trabajo en equipo.
- Elaborar en coordinación con el estudiante, una lista de palabras clave a investigar por unidad.
- Propiciar la búsqueda de información de la terminología de cada unidad.
- Solicitar el análisis del software libre existente en Internet y en la bibliografía de editores, compiladores(analizadores), ligadores y cargadores.
- Solicitar como proyecto final, un cargador, un ligador o un compilador de operaciones básicas (calculadora).

8.- SUGERENCIAS DE EVALUACIÓN

- Aplicar un examen inicial para conocer en nivel de los aprendizajes requeridos.
- Integrar al estudiante para establecer los porcentajes de cada punto.
- Participar en clase y laboratorio.
- Realizar trabajos de investigación en diversas fuentes de información.
- Elaborar reportes de prácticas.
- Exponer temas específicos en aula.
- Aplicar un examen final con la intención de verificar el logro de los objetivos educacionales del curso.

9.- UNIDADES DE APRENDIZAJE

UNIDAD 1.- Introducción.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
El estudiante conocerá la forma de trabajo de una computadora, y se concientizará del software de base que ya utiliza.	1.1 Investigar y redactar los conceptos de la guía proporcionada por el profesor. 1.2 Investigar y discutir en clase las fases existentes entre un problema y un programa cargado en memoria. 1.3 Identificar el editor en uso en cada uno de los paquetes instalados en su computadora. 1.4 Identificar el nivel, tipo y características de al menos 10 lenguajes en uso en la actualidad.	1, 2, 3

UNIDAD 2.- Ensamblador.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Conocerá la estructura de un programa, el formato de las instrucciones más comunes, el modo de direccionamiento y el uso de la memoria para comprender el funcionamiento del lenguaje.	<ul style="list-style-type: none">2.1 Investigar y redactar los conceptos de la guía proporcionada por el profesor.2.2 Investigar la estructura de un programa en ensamblador.2.3 Investigar el formato de las instrucciones más comunes en ensamblador.2.4 Modificar un programa ensamblador generado por el debugger para manejo de registros de máquina.2.5 Modificar un programa ensamblador generado por el debugger para manejar las operaciones básicas.2.6 Modificar un programa ensamblador generado por el debugger para manejar y usar la memoria.2.7 Modificar un programa ensamblador generado por el debugger para manejar macros.2.8 Modificar un programa ensamblador generado por el debugger para manejar interrupciones.	3, 4, 5, 6, 7

UNIDAD 3.- Compiladores.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Comprenderá la estructura interna de los compiladores e intérpretes para un uso más eficiente de los mismos.	<ul style="list-style-type: none">3.1 Investigar y redactar los conceptos de la guía proporcionada por el profesor.3.2 Investigar y discutir en clase las funciones de los analizadores y generadores de código de los compiladores.3.3 Diferenciar y concientizar las funciones de los diferentes analizadores.3.4 Desarrollar un traductor de interés del estudiante.	1, 2, 8, 10

UNIDAD 4.- Ligadores y cargadores.

Objetivo Educativo	Actividades de Aprendizaje	Fuentes de Información
Comprenderá el funcionamiento de los ligadores y cargadores estáticos y dinámicos.	4.1 Investigar y redactar los conceptos de la guía proporcionada por el profesor. 4.2 Investigar y discutir en clase las funciones de los ligadores estáticos y dinámicos. 4.3 Investigar y discutir en clase las funciones de los cargadores estáticos y dinámicos. 4.4 Desarrollar un cargador de interés del estudiante. 4.5 Concientizar el punto en el que se cede el control al núcleo del sistema operativo para la posterior ejecución del mismo.	1, 2, 9, 10

10. FUENTES DE INFORMACIÓN

1. Beck, Leland L.
Software de sistemas. Introducción a la programación de sistemas.
Ed. Addison Wesley Iberoamericana.
2. Donovan, Jhon, J.
Programación de sistemas.
Ed. . El ateneo.
3. Los microprocesadores Intel.
Brey, Barry B.
Ed. Prentice Hall.
4. Assembly Language and Systemms Programming for the IBM pc & compatible.
Lemone, Karen A.
5. IBM PC Assembly Language. A guide for programming.
Sacalon, Leo J.
6. Macroassembler for the MS DOS Operating Systems. Programmers Guide. Microsoft code view utilities update
Ed. Microsoft corporation.

7. Lenguaje ensamblador para IBM PC y compatibles.
Abel, Peter.
Ed. Prentice Hall.
8. Compiladores. Principios, técnicas y herramientas.
Alfred V. Aho, Ravi Sethi, Jeffrey D Ullman.
Ed. Addison Wesley Iberoamericana.
9. Assemblers, Linkers and Loadres.
Barrón, D. W.
Ed. North Holland.
10. Conceptos Fundamentales de Compiladores.
B. Teufel, S. Schmidt, T. Teufel.
Addison Wesley Iberoamericana.

11. PRÁCTICAS

Unidad Práctica

- 1 Identificar el editor en uso, en cada uno de los paquetes instalados en su computadora.
- 2 Identificar el nivel, tipo y características de al menos 10 lenguajes en uso en la actualidad.
- 3 Desarrollar o modificar programas en ensamblador generados por el debugger para conocer el manejo de los registros de máquina, el uso de las operaciones básicas, la forma de direccionamiento, el uso de macros y el uso de interrupciones.
- 4 Desarrollar un traductor de interés del estudiante.
- 5 Desarrollar un cargador de interés del estudiante.