

**UNIVERSIDAD AUTÓNOMA DE CHIAPAS  
LICENCIATURA EN SISTEMAS COMPUTACIONALES**

**Área de formación:** Disciplinaria  
**Unidad Académica:** Programación de Sistemas  
**Ubicación:** Cuarto Semestre  
**Clave:** 1984  
**Horas semana-mes:** 5  
**Horas teoría:** 3  
**Horas práctica:** 2  
**Unidades CONAIC:** 58.67  
**Prerrequisitos:** Ninguno  
**Horas de infraestructura:** 2  
**Créditos:** 8

### **PRESENTACIÓN**

El desarrollo de la programación de sistemas o software de base ha permitido que la comunicación entre la computadora y los seres humanos alcance un nivel superior con respecto del lenguaje de máquina, gracias al diseño de los traductores, específicamente los de bajo nivel, por lo que es de importancia para el estudiante conocer los conceptos relacionados con su funcionamiento, programación y diseño, para comprender la forma en que la computadora ejecuta las instrucciones.

### **OBJETIVO GENERAL**

Al concluir esta unidad académica el estudiante aplicará los conceptos relacionados con la programación de sistemas, sus elementos, la función, el uso y el diseño de los traductores de bajo nivel para lograr el ensamblado, carga, macroprocesamiento y ejecución de programas.

### **UNIDAD I.- INTRODUCCIÓN**

**TIEMPO APROXIMADO:** 10 Horas

**OBJETIVO DE LA UNIDAD:** Recuperar los conceptos básicos sobre la arquitectura interna de la computadora como base y parte importante de la programación de sistemas, recalcando la dependencia que existe entre hardware y el software.

### **CONTENIDO**

- 1.1 Programación de sistemas
- 1.2 Niveles de complejidad
- 1.3 Interrelaciones del sistema (Hardware y Software)
- 1.4 El modelo Von Newmann
- 1.5 Estructura del procesador
- 1.6 Lenguaje de máquina
- 1.7 Registros y modelo de programación
- 1.8 Memoria y modos de direccionamiento

### **UNIDAD II.- ELEMENTOS DE LA PROGRAMACIÓN DE SISTEMAS**

**TIEMPO APROXIMADO:** 10 Horas

**OBJETIVOS DE LA UNIDAD:**

- Comprender la diferencia que existe entre los traductores de bajo y alto nivel.
- Identificar los diferentes elementos de la programación de sistemas.

## **CONTENIDO**

- 2.1 Traductores de bajo y alto nivel
- 2.2 Ensambladores
- 2.3 Cargadores
- 2.4 Macroensambladores
- 2.5 Macroprocesadores

## **UNIDAD III.- ENSAMBLADORES**

**TIEMPO APROXIMADO:** 20 Horas

### **OBJETIVOS DE LA UNIDAD:**

- Experimentar los diferentes esquemas de ensamblado, así como las etapas necesarias para el diseño de un ensamblador.
- Programar en lenguaje ensamblador.

## **CONTENIDO**

- 3.1 Función de un ensamblador
- 3.2 Relación arquitectura de la máquina-ensamblador
- 3.3 Tipos de ensambladores
- 3.4 Procesos específicos para el proceso de ensamble
- 3.5 Modalidades y técnicas de ensamblado
- 3.6 Gestión de la memoria en el ensamblador
- 3.7 Diseño y programación del ensamblador

## **UNIDAD IV.- CARGADORES**

**TIEMPO APROXIMADO:** 10 Horas

**OBJETIVO DE LA UNIDAD:** Experimentar los diferentes esquemas de la liga de módulos objeto y rutinas de biblioteca.

## **CONTENIDO**

- 4.1 Función de un cargador
- 4.2 Tipos de cargadores
- 4.3 Relación cargador-sistema operativo
- 4.4 Funciones del editor de enlace
- 4.5 Liga de módulos objeto
- 4.6 Diseño y programación de un cargador

## **UNIDAD V.- MACROPROCESADORES**

**TIEMPO APROXIMADO:** 10 Horas

**OBJETIVO DE LA UNIDAD:** Interpretar el funcionamiento y los pasos necesarios para el diseño de un macroprocesador.

## **CONTENIDO**

- 5.1 Función y usos de un microprocesador
- 5.2 Macros: su importancia, definición y expansión simple, paramétrica y con anidamiento.
- 5.3 Tablas asociadas
- 5.4 Biblioteca de macros
- 5.5 Expansión condicional
- 5.6 Variables y operadores del proceso de expansión
- 5.7 Pasadas del texto fuente (preprocesamiento)
- 5.8 Macroprocesadores con argumentos y macroprocesadores recursivos
- 5.9 Incorporación de un procesador de macros a un ensamblador

## **UNIDAD VI.- MACROENSAMBLADORES**

**TIEMPO APROXIMADO:** 20 Horas

### **OBJETIVOS DE LA UNIDAD:**

- Interpretar las etapas que conforman a un macroensamblador.
- Aplicar los conceptos sobre ensamblado y ligado de programas utilizando un macroensamblador.

## **CONTENIDO**

- 6.1 Función de un macroensamblador
- 6.2 Definición de macroinstrucciones
- 6.3 Expansión de macroinstrucciones
- 6.4 Macroinstrucciones con parámetros
- 6.5 Ensamble condicional
- 6.6 Macrollamadas
- 6.7 Tablas y pasos del macroensamblador

### **EXPERIENCIAS DE APRENDIZAJE**

- Participación en foros de discusión
- Tareas de investigación individual y por equipo
- Lecturas
- Elaboración de resúmenes
- Prácticas de programación en laboratorio
- Elaboración de proyectos

### **PRÁCTICAS SUGERIDAS**

- Programas en ensamblador.
- Experimentos utilizando hardware para representación de datos.
- Realizar una investigación acerca de cómo se construye un microprocesador.
- Realizar una investigación acerca de los principales tipos de microprocesadores.
- Realizar un mapa mental para entender la interrelación entre hardware y software.
- Utilizar un emulador de microprocesador para entender su funcionamiento.

- Desarrollar programas básicos en ensamblador.
- Desarrollar programas básicos en C con ensamblador incrustado.
- Utilizar y configurar un cargador para estudiar su funcionamiento.
- Desarrollar programas que hagan uso de macroinstrucciones.
- Realizar experimentos utilizando hardware para representación de datos.
- Desarrollar una calculadora en lenguaje ensamblador que sume, reste, multiplique y divida números enteros.
- Desarrollar un traductor básico.

## CRITERIOS DE EVALUACIÓN

Exámenes	40%
Participación en foros de discusión	10%
Tareas y prácticas de laboratorio	20%
Proyecto final	30%
TOTAL	<u>100%</u>

## RECURSOS NECESARIOS

Recursos tecnológicos:  
TASM

### BIBLIOGRAFÍA BÁSICA

DETMER, R. (2009). Introduction to 80x86 Assembly Language and Computer Architecture. USA: Jones & Bartlett Publishers.  
 DURAN, L. (2007). El gran libro del PC interno. México: Alfaomega.  
 BREY, B. (2006). Microprocesadores Intel. México: Pearson Educación.  
 LEVINE, G. (2001). Computación y programación moderna. México: Pearson Educación.

### BIBLIOGRAFÍA COMPLEMENTARIA

BECK, L. (1997). System software. USA: Adison Wesley.  
 GODFREY, T. (1991). Lenguaje ensamblador para microcomputadores IBM para principantes y avanzados. México: Prentice Hall.