

UNIVERSIDAD AUTÓNOMA DE YUCATÁN
FACULTAD DE MATEMÁTICAS

MISIÓN

Formar profesionales altamente capacitados, desarrollar investigación y realizar actividades de extensión, en Matemáticas y Computación, así como en sus diversas aplicaciones.

LICENCIATURA EN CIENCIAS DE LA COMPUTACIÓN

OBJETIVOS:

1. Evaluar, seleccionar e implantar sistemas de información, así como organizar servicios y administrar recursos computacionales, con base en la estructura, operación y necesidades de información de las organizaciones.
2. Analizar, diseñar e implementar en forma multidisciplinaria, sistemas para administrar información útil en la toma de decisiones, mediante el uso de equipo computacional, diseñando métodos y procedimientos con apoyo de herramientas matemáticas y computacionales, que contribuyen a optimizar los recursos de las empresas.
3. Aplicar las metodologías actuales para el desarrollo general de software y de sistemas en particular, considerando entre otros: sistemas de información, aspectos de interconexión de equipos, interacción con sistemas operativos y teleprocesamiento.
4. Diseñar, construir, instalar y programar interfaces para aplicaciones en computación.

PROGRAMA: PROGRAMACIÓN DE SISTEMAS

NIVEL: SEPTIMO SEMESTRE

PROFESOR: ENRIQUE AYALA FRANCO

SEMESTRE: AGOSTO – ENERO

Autores del programa: Enrique Ayala Franco, Maximiliano Canché Euán, Fernando Curi Quintal, Jorge Gómez Montalvo.

Fecha de elaboración: 11 de septiembre de 2007.

Duración del curso: 75 horas.

Descripción de la asignatura:

En esta asignatura el estudiante profundizará en el funcionamiento de una computadora desde que se enciende hasta como se ejecutan los programas en un sistema multiusuario. Para lograr lo anterior, el estudiante analizará e implementará el funcionamiento de los principales componentes de diversos sistemas de base. Con la implementación de dichos componentes el estudiante mejorará sus habilidades de programación; así como también, utilizará las bibliotecas, utilerías, llamadas al sistema y comandos de los principales sistemas operativos.

Objetivo general:

- Utilizar la definición y métodos de construcción del software que hacen posible el funcionamiento de las computadoras en diferentes niveles de operación.
- Desarrollar software de base como: traductores, compiladores, herramientas y utilerías, DBMS, generadores de código, sistemas operativos, etc.

Contenido:

1. Conceptos Básicos de Programación de Sistemas (12 hrs.)

Objetivo: Identificar los tipos de sistemas de software que pueden ser desarrollados con apoyo de la teoría de Programación de Sistemas.

- 1.1. ¿Qué es la programación de sistemas?
- 1.2. Funcionamiento de un sistema de cómputo
- 1.3. Tipo de sistemas y sus características.
- 1.4. Metodologías y estándares para programación de sistemas
 - 1.4.1. POSIX
 - 1.4.2. Single UNIX Specification
 - 1.4.3. WinAPI32
- 1.5. Análisis de diversos sistemas abiertos

2. Software de Sistema Básico (15 hrs.)

Objetivo: Identificar las características del software de sistema básico en sus diferentes niveles de abstracción y su relación funcional con el Sistema Operativo.

- 2.1. Lenguaje Máquina
 - 2.1.1. Arquitecturas de computadoras
 - 2.1.2. Descripción de arquitecturas simplificadas
 - 2.1.3. Lenguajes máquina
- 2.2. Ensambladores
- 2.3. Lenguajes de programación
 - 2.3.1. Compiladores
 - 2.3.2. Intérpretes
- 2.4. Máquinas Virtuales

3. Sistemas Operativos (15 hrs)

Objetivo: Analizar y desarrollar algún subsistema de un sistema operativo.

- 3.1. Conceptos básicos de Sistemas Operativos
 - 3.1.1. Dispositivos de un sistema de cómputo, organización y estructura, procesos, archivos, seguridad,
 - 3.1.2. Interfaces: GUI, CLI, *shell de UNIX/Linux*.
- 3.2. Componentes principales de un Sistema Operativo
- 3.3. Subsistemas del Sistema Operativo
 - 3.3.1. Administración de procesos, Administración de Dispositivos de I/O, Administración de memoria, Administración de archivos.
- 3.4. Casos de estudio
 - 3.4.1. freeDos
 - 3.4.2. PCXINU
 - 3.4.3. Linux

3.4.4. Windows

3.4.5. MINIX

Prácticas sugeridas:

Desarrollar e implementar un subsistema del sistema operativo.

Desarrollo de un intérprete de comandos (shell).

4. Intérpretes y Compiladores (15 hrs.)

Objetivo: Diseñar e implementar un sistema que realice las funciones de un intérprete y/o compilador.

4.1. Intérpretes: características y componentes

4.2. Compiladores: características y componentes.

4.3. Diferencias entre intérpretes y compiladores.

4.4. Diseño e Implementación de los componentes de un compilador

4.4.1. Analizador léxico, sintáctico y semántico

4.4.2. Generación de Código objeto y ejecutable

Prácticas Sugeridas:

Desarrollo de una aplicación que realice las funciones básicas de un compilador y/o intérprete.

5. Software adicional del Sistema (15 hrs)

Objetivo: Desarrollar software de aplicación o herramientas complementarias que apoyan al sistema operativo.

5.1. Codificadores y Cifradores

5.2. Empaquetadores y respaldo de información

5.3. Procesadores de texto

5.4. Hojas de cálculo

5.5. Sistemas Manejadores de Bases de Datos (SMBD)

5.6. Monitoreo de actividades (dispositivos, red, CPU, colas de impresión)

5.7. Herramientas de uso común (Barras de acceso directo, reproductores de audio y video, calculadora, reloj, etc.)

Prácticas sugeridas:

Desarrollo de un editor de imágenes, de reproducción de información de multimedios, o un editor de texto.

Desarrollo de un manejador de bases de datos que permita leer diversos formatos de BDs y obtener información mediante un lenguaje de consultas prototipo (SQL resumido) y exportar a diversos formatos de BDs.

Bibliografía:

1. Beck, L. Software de Sistemas. Introducción a la Programación de Sistemas. Addison Wesley. 1990.

2. Tanenbaum, A. Operating Systems: Design and Implementation. 3ª. Ed. Prentice Hall.2006.
3. Tanenbaum, A. Organización de computadoras: un enfoque estructurado. Prentice Hall. 1988.
4. Aho, A. Ullman J. Principles of compiler design. Addison Wesley. 1977
5. Lister, A. Fundamentos de los sistemas operativos. Gustavo Gili 1986.
6. Kernigham B., Ritchie D. M. El lenguaje de programación C. Prentice Hall.

Metodología de Enseñanza:

- Conferencia, interrogatorio, resolución de ejercicios, tormenta de ideas, uso de software, trabajo en equipos, demostración, investigaciones bibliográficas.

Recursos didácticos:

Conferencia: diapositivas, acetatos, pizarrón.

Comisión (equipos): computadora.

Criterios de evaluación

Evaluaciones	Porcentaje	Duración	Fecha de aplicación
Prueba 1 (Unidades 1, 2)	20%	1 Sesión	14 de octubre
Prueba 2 (Unidad 3)	20%	1 Sesión	17 de noviembre
Prueba 3 (Unidad 4, 5)	20%	1 Sesión	20 de diciembre

Trabajos en el curso	Porcentaje
Tareas, Exposiciones	25%
Proyecto final	15%

Son requisitos para poder presentar el examen ordinario:

1. 80 % de asistencia en clase.
2. Haber entregado el 80% de las tareas asignadas y el proyecto final.
3. Haber presentado al menos dos exámenes parciales.

Recursos en Internet

Advanced Linux Programming
<http://www.advancedlinuxprogramming.com/>

The Single UNIX System Specification
<http://www.unix.org/version3/>

The MINIX 3 Operating System, 11-jul-2007
<http://www.minix3.org>

El proyecto FreeDOS, 11-jul-2007
<http://inicia.es/de/aitorsm/FreeDOS/FreeDOS.htm>

The DR-DOS-OpenDOS Enhancement Project, 11-jul-2007
<http://www.drDOSprojects.de/index.cgi/>

The Lex & Yacc Page
<http://dinosaur.compilertools.net>

A compact guide to lex & Yacc, Thomas Niemann
<http://epaperpress.com/lexandyacc>

Lex & Yacc HOWTO, Bert Hubert (PowerDNS.COM.BV)
http://ds9a.nl/lex_yacc

Bison
<http://www.gnu.org/software/bison/bison.html>

Flex
<http://www.gnu.org/software/flex/flex.html>

Compiler Construction using Flex and Bison, Anthony Aaby
<http://cs.wvc.edu/aabyan/464/Book/>

Flex, versión 2.5 A fast scanner generator, Edition 2.5, March 1995, Vern Paxson
http://www.cs.princeton.edu/appel/modern/c/software/flex/flex_toc.html

Bison, The Yacc-compatible Parser Generator, November 1995, Bison
Version 1.5, Charles Donnelly and Richard Stallman
http://www.cs.princeton.edu/appel/modern/c/software/bison/bison_toc.html

<http://www.docstoc.com/docs/2688078/Introduction-to-System-Software-the-Simplified-Instructional->

Código de SIC
<http://sicvm.cvs.sourceforge.net/viewvc/sicvm/sicvm/src/>

Curso Ensamblador y Programación de Sistemas

<http://140.115.52.150/~yu/course/AssemblySP/AssemblySP.htm>

Bases de datos en C

<http://elvex.ugr.es/software/dbms/spanish.html>