

PROGRAMA DE ESTUDIOS: PROGRAMACIÓN DE SISTEMAS

PROTOCOLO

Fechas	Mes/año
Elaboración	05-2010
Aprobación	
Aplicación	08-2010

Clave			Semestre	6°		
Nivel	Licenciatura	X	Maestría		Doctorado	
Ciclo	Integración		Básico		Superior	
Colegio	H. y C.S.		C. y T.	X	C. y H.	

Plan de estudios del que forma parte: Ingeniería de Software

Propósito(s) general(es):
 Que el estudiante identifique y describa la arquitectura y organización de un sistema computacional e implemente las diferentes técnicas empleadas en el proceso de generación de código para su uso y construcción en sistemas computacionales.

Carácter		Modalidad			Horas de estudio semestral (16 semanas)						
Indispensable	X	Seminario		Taller		Con Docente	Teóricas	36	Autónomas	Teóricas	20
		Curso	X	Curso-taller			Prácticas	36		Prácticas	36
Optativa *		Laboratorio	X	Clínica		Carga horaria semanal: 4.5 + 3.5 = 8		Carga horaria semestral:		128	

Asignaturas Previas:	Asignaturas Posteriores:
Construcción y evolución del software	Arquitectura de computadoras

Requerimientos para cursar la asignatura	Conocimientos y habilidades: Conocimientos de programación estructurada y modular, Programación Orientada a Objetos y Estructuras de Datos
---	---

Perfil deseable del profesor:	Ingeniero de Software, Licenciado en Ciencias Computacionales, Ingeniero en Computación o afín, preferentemente con postgrado y experiencia docente, de investigación o trabajo en el área.
--------------------------------------	---

Academia responsable del programa:	Diseñador (es):
Academia de Informática	Omar Nieto Crisóstomo

* Aquellas en las que se ofrece la posibilidad de cursar una de las asignaturas, para cubrir un requisito INDISPENSABLE será considerada INDISPENSABLE.

INTRODUCCIÓN

Hoy en día los sistemas computacionales se utilizan, cada más, en dispositivos de uso común, completamente diferentes a los de una computadora personal, por ejemplo en televisores, lavadoras, refrigeradores, radio, automóviles, celulares, etc. Cada uno de estos dispositivos, están gestionados por una unidad de procesamiento central, de tal manera que, esta última, coordina el funcionamiento del aparato para tener un rendimiento óptimo. La materia de Programación de Sistemas introduce al estudiante a la arquitectura y organización de un sistema computacional para que programe a bajo nivel una arquitectura de cómputo y construya herramientas básicas de generación de código para su implantación y construcción de sistemas computacionales. El primer capítulo muestra la forma de representar la información en un sistema de cómputo, la conversión de bases numéricas (binario, octal y hexadecimal) y la manera de realizar operaciones binarias. El capítulo segundo introduce al estudiante a la organización y arquitectura de un sistema de cómputo, los elementos que constituye la unidad de procesamiento central, la representación de la memoria principal y el acceso a puerto de entrada y salida. El capítulo tercero, incluye los conceptos de programación de un sistema de cómputo a bajo nivel, la manera de creación de programas básicos empleando instrucciones propias del procesador en ensamblador y la manera de interpretarlas en código máquina. El cuarto y quinto capítulo introduce las técnicas empleadas para la creación de herramientas para la generación de código, en el cuarto capítulo se muestran las técnicas para la generación de código a nivel ensamblador y el quinto capítulo se muestra algunas técnicas empleadas para la generación de código mediante lenguajes de alto nivel. La materia de Programación de Sistemas se complementa con las asignaturas de Introducción a la Programación y Programación Orientada a Objetos para el mejor entendimiento de la Programación de sistemas computacionales, es necesario haber acreditado esta materia par tener éxito en las materias de Sistemas Operativos y Arquitectura de Computadoras.

PROPÓSITOS GENERALES

Que el estudiante identifique y describa la arquitectura y organización de un sistema computacional e implemente las diferentes técnicas empleadas en el proceso de generación de código para su uso y construcción en sistemas computacionales.

CONTENIDOS

TEMAS Y SUBTEMAS	PROPÓSITOS ESPECIFICOS
1. Introducción a la programación de sistemas	Que el estudiante identifique y describa los niveles de abstracción en la programación de sistemas cómputo
1.1 Niveles de abstracción de programación de sistemas	Que el estudiante maneje las diferentes bases numéricas, sus conversiones y operaciones que se pueden realizar, así como la representación de la información en un sistema de cómputo.
1.2 Bases numéricas (binario, octal, hexadecimal)	
1.3 Representación de números enteros con signo, números de punto fijo, números de punto flotante, caracteres y cadenas	
1.4 Operaciones binarias (suma, resta, multiplicación, AND, NOT, OR)	

<p>2. Arquitectura de la computadora</p> <p>2.1 Unidad de procesamiento central (ALU, Unidad de control, Registros)</p> <p>2.2 Memoria principal</p> <p>2.3 Puertos de E/S e interrupciones</p> <p>2.4 Maquina de Von Neumann (Ciclo de Fetch)</p>	<p>Que el estudiante identifique y describa los elementos que conforma a la unidad de procesamiento central (CPU), la manera como se accede la memoria principal y a los puertos de entrada y salida, finalmente, que describa el proceso general que realiza el CPU al ejecutar una instrucción.</p>
<p>3. Lenguaje ensamblador</p> <p>3.1 Proceso de generación de código</p> <p>3.2 Código a nivel de máquina</p> <p>3.3 Modos de direccionamiento (maquinas con cero, uno, dos y tres modos de direccionamiento)</p> <p>3.4 Uso de etiquetas para código y datos</p> <p>3.5 Instrucciones de lectura y almacenamiento a memoria</p> <p>3.6 Instrucciones aritméticas y lógicas</p> <p>3.7 Instrucciones de control</p> <p>3.8 Llamadas a subrutina</p> <p>3.9 Invocar rutinas en ensamblador desde un lenguaje de alto nivel</p>	<p>Que el estudiante use instrucciones básicas de ensamblador de una arquitectura de cómputo, para que realice programas sencillos y pueda traducirlos a código a nivel de máquina.</p> <p>Que el estudiante identifique los elementos que intervienen en la elaboración de programas a nivel ensamblador, mediante el uso de un programa traductor de ensamblador a código máquina.</p>
<p>4. Ensambladores, Cargadores y Ligadores</p> <p>4.1 Ensambladores de una pasada y de dos-pasadas</p> <p>4.2 Direccionamiento absoluto y relativo</p> <p>4.3 Segmentación y paginación de memoria</p> <p>4.4 Directivas</p> <p>4.5 Macros</p> <p>4.6 Cargadores</p> <p>4.7 Bootstrap</p>	<p>Que el estudiante identifique y describa los elementos que intervienen en la generación de código: ensamblador, ligador y cargador, para que el estudiante identifique las características necesarias a considerar, en la elaboración de herramientas básicas de generar código de manera independiente.</p>
<p>5. Introducción a los traductores de lenguajes</p> <p>5.1 Lenguajes, gramáticas y analizadores (parsing)</p> <p>5.2 Maquinas de estado finito</p> <p>5.3 Implantación de Máquinas de estado finito</p> <p>5.4 Generación de código</p>	<p>Que el estudiante identifique y describa los elementos básicos a considerar en la programación de traductores de lenguajes, para que el estudiante programe un traductor de lenguaje ensamblador a lenguaje máquina.</p>

<p>6. Introducción a los compiladores</p>	<p>Que el estudiante identifique y describa los módulos básicos que intervienen en la construcción de programas compiladores.</p>
<p>6.1 Compiladores e interpretes 6.2 Analizador Léxico 6.3 Analizador sintáctico 6.4 Analizador semántico 6.5 Optimizador de código 6.6 Generador de código</p>	

METODOLOGÍA PARA EL CURSO

La materia se impartirá con clases teóricas y prácticas de laboratorio. El curso se llevará a cabo mediante exposiciones orales, resolución de ejercicios, trabajos en equipo, trabajos de investigación, asesorías, prácticas de laboratorio en donde se practique y verifique el buen funcionamiento de los algoritmos. Es preferente que el estudiante realice un proyecto final en donde se aplique varias de las técnicas vistas en clase.

EVALUACIÓN DIAGNÓSTICA

Se aplicará un examen escrito donde se indague los conocimientos de: programación estructurada y modular, matemáticas discretas y fundamentos de programación orientada a objetos. Esta evaluación le permite al docente, identificar los conocimientos, habilidades y deficiencias que poseen los estudiantes, de tal manera que, pueda realizar acciones que sean orientadas al mejor aprovechamiento del estudiante en la materia, en base a los resultados de la evaluación, el profesor podrá realizar repases, programar asesorías, proponer investigaciones en bibliografía, así como, identificar metodología convenientes en la enseñanza y el aprendizaje.

EVALUACIÓN FORMATIVA

Se recomienda al docente que aplique mínimamente tres evaluaciones formativas a lo largo del semestre, en donde el estudiante demuestre los conocimientos y habilidades adquiridas durante dicho periodo de evaluación. En la parte práctica, se recomienda que el profesor deje un trabajo de investigación previo a cada práctica, referente al tema a tratar, y que el estudiante entregue el reporte de la práctica. Además, al final del semestre, se recomienda que el estudiante entregará un trabajo escrito con el reporte técnico de la elaboración de un proyecto final donde aplique las técnicas vistas en clase.

EVALUACIÓN DE CERTIFICACIÓN

El examen de certificación evaluará los conocimientos y habilidades del estudiante adquiridos durante el curso. Este instrumento consta de dos partes: una evaluación teórica, y una evaluación práctica. La evaluación teórica consiste en la elaboración de un examen escrito en donde se evaluarán los conceptos, definiciones, las técnicas de programación de sistemas, su uso apropiado y las aplicaciones de estas técnicas en problemas específicos. La segunda parte consta de un proyecto, donde el estudiante aplique varias técnicas vistas en clase en un problema específico, el estudiante entregará un reporte técnico que describa los detalles en todo del proceso de desarrollo de software de su proyecto, también entregará los archivos fuente y ejecutable en un dispositivo de almacenamiento.

BIBLIOGRAFÍA

La bibliografía que se menciona a continuación es la básica para el estudiante y el profesor:

Bibliografía	Temas para los que se recomienda
1. Beck, Leland, System An Software: Introduction to Systems Programming. 3er Edition. Addison Wesley. 1997	Todos
2. Bryant, Randal E. and O'Hallaron, David R. Computer Systems A programmer's perspective. Prentice Hall. 2003.	1, 2 y 3
3. 5. Michael Tischer, Bruno Jennrich. PC Intern: The Encyclopedia of System Programming. Abacus Software, 1996	1, 2 y 3
4. Warford J. Stanley. Computer Systems, 2a Edición, Jones And Bartlett Publishers, 2002	Todos
5. Salomon David, Assemblers And Loaders, Ellis Horwood Ltd, 1993	4 y 5
6. Kenneth Loudon, Construcción de compiladores principios y practica, Thomson, 2004	6
7. Calingaert Peter, Assemblers, Compilers, and Program Translation, Computer Science Press, 1979	4,5, y 6
8. Silvarama P. Dandamudi, Introduction to Assembly Language Programming, For Pentium and RISC Processors. Springer, 2004	1, 2, 3 y 4
9. Sivarama P. Dandamudi, Guide To Assembly Language Programming in Linux, Springer, 2005	1, 2 y 3
10. Neveln Bob, Linux Assembly Language Programming, Prentice Hall, 2000	1, 2 y 3

OTROS RECURSOS

Proyector de video (Cañon), Computadora que tenga instalado un compilador, ensamblador, depurador, ligador, cargador y editor de texto; de preferencia que se encuentre instalado un entorno de desarrollo integrado que contenga todos los programas previos. Se recomienda que se cuente con una arquitectura de cómputo donde se practique lo visto en clase.