LENGUAJES DE PROGRAMACIÓN Y SUS PARADIGMAS

CLAVE: 0607	ÁREA: CIENCIAS DE LA COMPUTACIÓN	
SEMESTRE: VI	Requisitos: Teoría de la Computación	
CRÉDITOS: 10		
HORAS POR CLASE	TEÓRICA: 1	TEÓRICO-PRÁCTICAS: 2
CLASES POR SEMANA	TEÓRICA: 4	TEÓRICO-PRÁCTICAS: 1
HORAS POR SEMESTRE	TEÓRICA: 64	TEÓRICO-PRÁCTICAS: 32

Objetivos generales:

Este curso introduce al estudiante a la naturaleza de los lenguajes de programación contemporáneos, empezando por una revisión profunda de lenguaje ensamblador y finalizando con un tratamiento profundo de un lenguaje verdadero orientado a objetos (tal como Smalltalk o Eiffel). Durante el desarrollo del curso se discute la evolución de lenguajes imperativos (FORTRAN, Algol, PL/I, Pascal) y lenguajes funcionales (Lisp, Scheme, ML). Adicionalmente se introducen conceptos fundamentales de diseño e implementación de los lenguajes de alto nivel, incluyendo los conceptos de ligado, chequeo de tipos y administración de memoria durante ejecución.

Temario:

I. Historia y evolución de los lenguajes de programación 2 horas

Una revisión histórica breve de los principales desarrollos en los lenguajes de programación, empezando por los lenguajes de alto nivel orientados a procedimientos. Panorama de los paradigmas de programación contemporáneos y los lenguajes asociados a ellos, incluyendo orientados a procedimientos, funcionales, orientados a objetos y programación paralela

- I.1 Primeros lenguajes: Algol, Fortran, Cobol
- I.2 La evolución de los lenguajes orientados a procedimientos (las cadenas de desarrollo de Algol, PL/I, Pascal, Euclid, Modula y Ada)
- I.3 Paradigmas y lenguajes no orientados a procedimientos: funcionales (LISP), lógicos (Prolog), orientados a objetos (Smalltalk) y paralelos (Occam)

II. Organización de la computadora a nivel de ensamblador 8 horas

Comparación de distintos juegos de instrucciones y sus correspondientes modos de direccionamiento. Énfasis en la interrelación entre conjuntos de instrucciones, operaciones de carga y ejecución y la arquitectura subyacente. Introducción al concepto de interrupciones, así como al propósito y especificaciones de una unidad de control con respecto a las operaciones lógicas. Unidades de control alambradas y microprogramadas, sus ventajas y desventajas respectivas. Microcódigo vertical y horizontal. Métodos generales para diseñar para el mantenimiento, tales como modularizar el diseño para facilitar el mismo o agregar unidades de hardware para facilitar el acceso a ciertos registros.

II.1 Organización básica: Von Neumann, diagramas de bloque, rutas para datos, rutas de control, unidades funcionales (i.e. Unidad Aritmético-lógica, memoria, registros), ciclo de instrucción
II.2 Conjuntos y tipos de instrucciones
II.3 Lenguaje ensamblador/de máquina
II.4 Modos de direccionamiento (i.e. directo, indirecto, de desplazamiento, de registro, indexamiento)
II.5 Unidad de control; carga y ejecución de la instrucción; carga del operando
II.6 Interrupciones de entrada/salida
II.7 Instrumentación por alambrado
II.8 Instrumentación por microprogramación; formatos y codificación.

III. Máquinas virtuales 2 horas

Contraposición de máquinas reales con máquinas virtuales. La comprensión de los lenguajes de programación en términos de las máquinas virtuales subyacentes a cada uno de ellos (independientemente de la arquitectura real en la que corren los programas escritos en esos lenguajes). La traducción del lenguaje en términos de la implementación en una máquina virtual, seguida de una sucesión de traducciones a través de una jerarquía de máquinas virtuales. Momentos de ligado como una noción importante para entender la semántica de los lenguajes.

III.1 Máquinas virtuales para los lenguajes de programación
III.2 Jerarquía de máquinas virtuales, presentadas al usuario a través del programa, el traductor, el sistema operativo, etc.
III.3 Consecuencias para la traducción de los distintos momentos en los que se hace el ligado

IV. Control de secuencia 4 horas

Flujo del control en los lenguajes de programación para la evaluación de expresiones y la ejecución de enunciados. Expresiones y enunciados definidos por el usuario

IV.1 Expresiones, orden en la evaluación, efectos laterales.
IV.2 Enunciados: simples y compuestos
IV.3 Subprogramas y corrutinas como una abstracción de expresiones y enunciados

V. Control de los datos, maneras de compartirlos, chequeos de tipo 8 horas

Métodos para compartir datos y restricción del acceso a datos en los lenguajes de programación. Chequeo e inferencia de tipos

V.1 Mecanismos para compartir y restringir el acceso a datos (estructura de bloques, COMMON, ADTs y aliases)
V.2 Rangos estático vs. dinámico, extensión, visibilidad
V.3 Mecanismos para el paso de parámetros: Por referencia, por valor, por nombre, por resultado, etc.
V.4 Variedad en las disciplinas para el chequeo de tipos y sus mecánicas; estática vs. dinámica vs. sin tipo, explícita vs. implícita, polimorfismo vs. sobrecarga

VI. Manejo del espacio de almacenamiento durante ejecución 8 horas

Asignación, recuperación y reutilización del espacio de almacenamiento durante la ejecución del programa.

VI.1 Asignación estática
VI.2 Asignación basada en un stack y su relación con la recursividad
VI.3 Asignación basada en una estructura de heap

VII. Paradigmas de programación 12 horas

Introducción a distintos paradigmas de programación alternativos (funcional, lógico y orientado a objeto) y sus lenguajes (LISP, Prolog y Smalltalk respectivamente). Construcción de programas utilizando al menos dos de estos paradigmas. Ventajas y desventajas frente al paradigma de programación orientada a procedimientos

VII.1 Revisión de los paradigmas y lenguajes funcional, lógico y orientado a objetos
VII.2 Diseñar programas con estos paradigmas; ambiente de ejecución, flujo de control
VII.3 Programas ejemplo y aplicaciones
VII.4 Ventajas (transparencia en la referencia) y desventajas (eficiencia en arquitecturas secuenciales) de los paradigmas de programación alternativos frente a los lenguajes orientados a procedimientos; aplicaciones en inteligencia artificial, bases de datos y diseño de sistemas de computación

VIII. Diseño de Lenguajes: Semántica 8 horas

Se da una visión comprensiva de la semántica de los lenguajes de programación, identificando los principales aspectos así como sus soluciones, en las implementaciones de los lenguajes de programación contemporáneos, revisando fundamentalmente los conceptos de tipos, asignación de memoria, estructuras de control, procedimientos y parámetros y, finalmente, entornos en tiempo de ejecución.

VIII.1 Una máquina sencilla y el modelo denotacional.
VIII.2 Tipos, vinculación, operadores y coerción.
VIII.3 Asignación de memoria.
VIII.4 Estructuras de control.
VIII.5 Procedimientos y parámetros.

IX. Diseño de Lenguajes: Pragmática 8 horas

Se identifican las principales fuerzas que influyen actualmente en la incorporación de nuevas características a los lenguajes de programación y la disolución de las más viejas. Se concluye con una evaluación comparativa de los lenguajes presentados en el curso, junto con algunas observaciones sobre el proceso mismo de evaluación de lenguajes.

IX.1 El arte y ciencia del diseño de lenguajes.
IX.2 El arte y ciencia de la programación.
IX.3 Entorno de programación.
IX.4 Comparación y evaluación de lenguajes.

IX.5 Conclusiones.

Bibliografía:

Básica:

- Friedman, D. P.; Wand, M.; Heynes, C. T., ***Essentials Of Programming Languages***, The MIT Press, 1992
- Sethi, R., ***Programming Languages, Concepts and Constructs***, Addison-Wesley Publishing Company, 1989

Complementaria:

- Scragg, G. W., ***Computer Organization, A Top-Down Approach***, McGraw-Hill Publishing Company, Inc., 1992
- Budd, T., ***An Introduction To Object-Oriented Programming***, Addison-Wesley Publishing Company, 1991
- Field, A. J.; Harrison, P. G., ***Functional Programming***, Addison-Wesley Publishing Company, 1989
- Friedman, L. W., ***Comparative Programming Languages, Generalizing The Programming Function***, Prentice Hall, Inc., 1991
- Kogge, P. M., ***The Architecture of Symbolic Computers***, McGraw-Hill Incorporated, 1991
- Tucker, A. B., Jr., ***Lenguajes De Programación, Segunda Edición***, McGraw-Hill,, España, 1987.