

PROGRAMA DE CURSO

CÓDIGO	NOMBRE DEL CURSO		
CC31A	Programación de Software de Sistemas		
NÚMERO DE UNIDADES DOCENTES	HORAS DE CÁTEDRA	HORAS DE DOCENCIA AUXILIAR	HORAS DE TRABAJO PERSONAL
10	3	1,5	5,5
REQUISITOS	REQUISITOS DE CONTENIDOS ESPECÍFICOS	CARÁCTER DEL CURSO	
CC30A	Computación <ul style="list-style-type: none"> • Programación Imperativa • Diseño de Algoritmos • Estructuras de Datos 	Obligatorio para Ingeniería Civil en Computación	
PROPÓSITO DEL CURSO			
Este curso busca que los alumnos aprendan a programar en lenguaje C, utilizando las funciones básicas que provee el Sistema Operativo Unix tanto para la administración de sistemas, para hacer programas eficientes y para programación concurrente. Se entregan los conceptos básicos de arquitectura de computadores como direcciones de memoria notación hexadecimal, representación de enteros, bits, etc y se enfatizan las funciones de manejo de memoria, Entrada/Salida, sistema de archivos, sockets y threads.			
OBJETIVO GENERAL			
Al finalizar el curso el alumno será capaz de: escribir y entender programas eficientes en lenguaje de programación C, manejar los conceptos básicos de la programación de sistemas y poder desarrollar sistemas para el Sistema Operativo Unix.			

UNIDADES TEMÁTICAS

NÚMERO	NOMBRE DE LA UNIDAD	OBJETIVOS
1	Programación Eficiente en C	Aprender a programar eficientemente en C y mostrar sus diferencias con los lenguajes de alto nivel (como Java)
DURACIÓN		
6 sem.		
CONTENIDOS		BIBLIOGRAFÍA
1.1 Lenguaje C (tipos, punteros, cast) 1.2 Representación de los datos (enteros, bits, hexadecimal) 1.3 Manejo de memoria (global, local, dinamica) 1.4 Strings, arreglos, punteros 1.5 Estructuras (struct) 1.6 Profiling y debugging 1.7 setjmp/longjmp 1.8 Macros, switch, tabla de saltos		[Kernighan]

NÚMERO	NOMBRE DE LA UNIDAD	OBJETIVOS
2	Sistema Operativo Unix	Aprender a usar eficientemente desde C las interfaces que provee el S.O. Unix.
DURACIÓN		
6 sem.		
CONTENIDOS		BIBLIOGRAFÍA
2.1 Entrada/Salida (open,read,write,lseek,close), File Descriptors, Sistema de Archivos, Directorios 2.2 Señales 2.3 Procesos (fork/exec), redirección de entrada/salida 2.4 Pipes, FIFOs 2.5 Manejo de Dispositivos (ioctl) 2.6 Sockets, Redes, Servidores multi-clientes (select)		[Stones, Cap. 1, 3, 10, 12, 14]

NÚMERO	NOMBRE DE LA UNIDAD	OBJETIVOS
3	Programación Concurrente	Aprender a desarrollar programas concurrentes correctos en C (threads)
DURACIÓN		
2 sem.		
CONTENIDOS		BIBLIOGRAFÍA
3.1 Threads y diferencias con procesos pesados 3.2 Sincronización: exclusión mutua, semáforos, monitores 3.3 Ejemplos clásicos: productos/consumidor, filósofos 3.4 Servidores de red multi-threads		[Nichols] [Silberschatz, cap. 6]

NÚMERO	NOMBRE DE LA UNIDAD	OBJETIVOS
4	Herramientas para la Programación de sistemas	Conocer otras herramientas disponibles además de C
DURACIÓN		
2 sem.		
CONTENIDOS		BIBLIOGRAFÍA
4.1 Shell Programming 4.2 Perl Programming 4.3 Web Programming (cgi-bin, php)		[Stones, Cap. 2, 18, 19]

BIBLIOGRAFÍA	EVALUACIÓN	
[Kernighan] Kernighan, B y Ritchie, D (1988) "The C Programming Language", Prentice-Hall, ISBN 0-13-110362-8	La evaluación se basa en tres controles y un examen (con apuntes y libros) más varias (entre 5 y 7) tareas de programación que son incrementales (se requiere usar programas de tareas anteriores para las siguientes) y que deben funcionar correctamente. Se sigue la ponderación que se plantea a continuación: $NC = (C1+C2+C3+Ex)/4$ $NT = (NT1+... NTn)/n$ $NF = 0,6*NC + 0,4*NT$	
[Stones] Richard Stones, Neil Matthew (2003), "Beginning Linux Programming (Programmer to Programmer)", Wiley, ISBN: 0-7645-4373-3		
[Nichols] B. Nichols, D. Buttlar, J. Proulx (1996) "Pthreads Programming: A POSIX Standard for Better Multiprocessing", O'Reilly, ISBN: 1-56592-115-1		
[Silberschatz] A Silberschatz et al, (2004) "Operating System Concepts", Wiley, ISBN: 0471694665		
FECHA DE VIGENCIA	ELABORADO POR	REVISADO POR
Otoño 2005	José M. Piquer	Luis Mateu