



CÓDIGO ASIGNATURA
0625

DEPARTAMENTO: *Ingeniería e Investigaciones Tecnológicas*

ASIGNATURA: Sistemas Operativos

Ingeniería en Informática
Año: 4º Cuatrimestral

2.2.- PROGRAMA ANALÍTICO.

CONTENIDOS TEÓRICOS Y PRÁCTICOS DE LA ASIGNATURA: SISTEMAS OPERATIVOS – VERSIÓN 2012

UNIDAD TEMATICA 1: Estructura de un sistema operativo

- 1.1. El arranque de la máquina. El sistema de booteo. COOL START o inicio en frío. Warm Start o inicio tibio. Inicialización del bootstrapping. Inicialización vía software. La secuencia del boot. Tipos según el sistema operativo.
- 1.2. El cargador de arranque. Flujo del programa cargador. El archivo IO.SYS. El POST.
- 1.3. El kernel del sistema operativo. Los diferentes tipos. El cargador del sistema operativo.
- 1.4. El modo real y el modo protegido. Diferencias entre los dos modos. La compuerta A20.
- 1.5. El sistema operativo SODIUM. Su desarrollo. Partes esenciales.
- 1.6. La instalación con multibootloader. El SOLO (SODIUM LOADER) estructura y funcionamiento.

Objetivo de la Unidad: Describir conceptos básicos que debe manejar el alumno para poder comprender el funcionamiento del sistema operativo SODIUM que será la base del desarrollo que ellos generarán durante el corriente año con el fin de ampliarlo y pulir los posibles errores que el mismo pueda tener.

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Realizar modificaciones al sistema operativo en desarrollo
- Explicar y reconocer las distintas partes que lo componen
- Operar el sistema operativo para parametrizarlo.

UNIDAD TEMATICA 2: Registros del procesador

- 2.1. Concepto de registro. Diferentes tipos y sus utilidades
- 1.7. Los registros reservados y su uso para la construcción de un sistema operativo. Los registros y su comportamiento. Registros de propósito múltiples. Registros de segmento. Puntero de instrucciones. Registros de direcciones del sistema. Contenidos probables y necesarios. Registros de Control, Registros de depuración y verificación, Registros de Control
- 2.3. Los registros ocultos. La función del microprocesador. Intel (porque en los laboratorios solo existen máquinas con esa tecnología y en sus domicilios también) la manera de cargar los registros ocultos
- 2.4. Los registros FLAG. Su uso y la necesidad de chequeo por parte del sistema operativo.
- 2.5. Los registros de la GDT, LDT e IDT

Objetivo de la Unidad: Describir las distintas formas de registros que se encuentran en un procesador. Entender que los registros que se consideran reservados ahora están a su alcance desde el punto de vista del sistema operativo

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Comprender y explicar las distintas formas de registros.
- Crear y ejecutar operaciones utilizando registros.
- Conocer y explicar la terminología específica empleada en este módulo.

UNIDAD TEMATICA 3: Assembler



- 3.1. Requerimientos para el uso del lenguaje assembler. Estructura del lenguaje assembler. Software necesario. Formato interno y externo de un programa.
- 3.2. Distintos tipos de lenguaje assembler. El MASM y el NASM. Su necesidad y la historia de su creación. Uso del enlazador.
- 3.3. El flujo de datos con los registros. Operaciones lógicas y aritméticas..
- 3.4. Condiciones, ciclos y bifurcaciones.
- 3.5. Las instrucciones SHORT y LONG su comportamiento.
- 3.6. Las instrucciones RET e IRET su comportamiento y su relación con los registros.

Objetivo de la Unidad: Describir las bases del lenguaje assembler.

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Explicar y reconocer la estructura del lenguaje assembler.
- El comportamiento de las instrucciones
- Realizar programas que relacionen el lenguaje con el sistema operativo.

UNIDAD TEMATICA 4: Relación entre el lenguaje assembler y el lenguaje C.

- 4.1. ¿Porqué mezclar ambos lenguajes? La necesidad de relacionar los lenguajes
- 4.2. Ventajas de uno respecto del otro.
- 4.3. Archivos C y assembler compilados individualmente.
- 4.4. Etapas de la compilación. Compilación del lenguaje C embebiendo el lenguaje assembler.
- 4.5. Las convenciones de llamadas a las funciones que se deben respetar para obtener un programa sin errores. Vinculación de archivos a través de scripts, makefiles, variables de entorno

Objetivo de la Unidad: Diseñar el modelo más favorable a aplicar para el desarrollo de programas. Determinar la conveniencia de realizar programas separados y unirlos por llamadores. Los movimientos del STACK. Las variaciones automáticas y los apuntadores especiales.

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Explicar y reconocer los distintos tipos de lenguajes assembler.
- Identificar la relación entre los lenguajes a través de las convenciones establecidas.
- Realizar programas que relacionen ambos lenguajes.
- Conocer la jerarquía de datos almacenados en un STACK y las operaciones que se pueden hacer entre ellos
- Conocer y explicar la terminología específica empleada en este módulo.

UNIDAD TEMATICA 5: El booteo y la estructura del disco en memoria

- 5.1. El boot y los mensajes de error. Causas por las que se provocan.
- 5.2. La estructura del disco en la memoria. Importancia. Aceleración de la transferencia de datos. Cuidados que se deben tener en consideración.
- 5.3. Introducción al MBR. Las diferentes estructuras. Las autorizaciones.

Objetivo de la Unidad: Describir qué se entiende por organización del booteo en una computadora, los distintos tipos que existen, la forma en que se representan los diferentes tipos de estructuras, las principales operaciones que realizan.

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Explicar y reconocer los distintos tipos de boot que existen.
- El funcionamiento del MBR y la estructura.
- Los programas almacenados.
- Conocer y explicar la terminología específica empleada en este módulo.

UNIDAD TEMATICA 6: Las interrupciones y las excepciones.

- 6.1. Los distintos tipos. Interrupciones de HARDWARE. Interrupciones de SOFTWARE.
- 6.2. Interrupciones enmascarables y no enmascarables. Su comportamiento. Aplicaciones con los registros intervinientes



6.3. Excepciones del procesador. Fault, traps y aborts. Las condiciones. Su relación con los registros del procesador. Condiciones que generan las interrupciones y las excepciones.

6.4. La relación entre las interrupciones de software y de hardware

6.5. La construcción de la IDT. El descriptor de la IDT. Procedimientos.

Objetivo de la Unidad: Describir qué se entiende por interrupciones, los distintos tipos que existen, la forma en que se relacionan. Describir que se entiende por excepciones, las herramientas que dichas técnicas utilizan. Describir el ciclo de vida de las mismas.

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Explicar y reconocer los distintos tipos de interrupciones.
- Explicar y reconocer los distintos tipos de excepciones.
- Identificar las diferentes formas de mostrar las relaciones entre el hardware y el software.
- Conocer y explicar la terminología específica empleada en este módulo.

UNIDAD TEMATICA 7: Master boot record

7.1. Como trabaja el MBR. Sumario de cosas que hace. Decodificación de una MBR. La tabla de particiones. El pequeño programa que se aloja en el MBR.

7.2. Particiones primarias, extendidas y lógicas. Volumen del disco o diskette.

7.3. Particiones activas y boot manager. Como bootea del DOS.

7.4. Virus en el boot sector.

7.5. Introducción a las llamadas al sistema.

7.6. El sistema de archivos. Introducción.

Objetivo de la Unidad: Describir qué se entiende por MBR. Describir como se trabajan las particiones de un disco. Describir como se instala el arranque en un diskette. Entender la importancia de las llamadas al sistema

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Explicar y reconocer al MBR.
- Explicar el funcionamiento y los distintos tipos de particiones existentes.
- Conocer y explicar la terminología específica empleada en este módulo.

UNIDAD TEMATICA 8: Las llamadas al sistema.

8.1. Tipos de servicios que ofrecen.

8.2. System calls para el manejo de procesos y su interrelación con la memoria y la CPU.

8.3. Relación entre el sistema usuario y el servicio que presta el hardware.

8.4. Ventajas de la capa intermedia de las llamadas al sistema. Facilitación de la programación.

8.5. Las llamadas al sistema y las API's. La función fork(). La función exit().

8.6. El proceso de una llamada. La biblioteca libc. La memoria de un proceso y las systems calls relacionadas. El llamado al Kernel. El paso de parámetros. La salida del Kernel.

8.7. Las llamadas al sistema bajo el estándar POSIX.

Objetivo de la Unidad: Describir qué se entiende por llamadas al sistema, los distintos tipos que existen, sus componentes, características, parámetros utilizados, normas más difundidas. Describir conceptos básicos para la construcción de llamadas al sistema.

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Explicar y reconocer los distintos tipos de llamadas al sistema.
- Conocer las características de las llamadas al sistema.
- Comprender y explicar las relaciones
- Construir y aplicar llamadas al sistema para un sistema operativo didáctico.

UNIDAD TEMATICA 9: La memoria

9.1. El papel del administrador de memoria. Características fundamentales del manejo de memoria.

9.2. Los administradores: Agrupamientos. Los administradores contiguos y no contiguos.

9.3. El direccionamiento de la memoria. Concepto de dirección lógica, lineal y física.

9.4. Administración de memoria simple. Características generales.



- 9.5. Administración de memoria contigua. Características generales.
- 9.6. Administración de memoria no contigua. Características generales.

Objetivo de la Unidad: Entender las variaciones de la administración de memoria, los distintos tipos que existen, sus componentes, características, parámetros utilizados, normas más difundidas. Describir conceptos básicos para la construcción de administradores de memoria físicos y virtuales.

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Explicar y reconocer los diferentes administradores de memoria.
- Conocer las características de cada uno de ellos
- Comprender y explicar las diferentes formas que tiene la unidad de memoria llamada segmento
- Programar un administrador de memoria para un sistema operativo didáctico

UNIDAD TEMATICA 10: El sistema de archivos.

- 10.1. Introducción. Función general del sistema de archivos. El concepto de archivo. Atributos del sistema de archivos. Variaciones entre los diferentes sistemas operativos.
- 10.2. Operaciones con archivos. Las llamadas involucradas. La importancia de las llamadas al sistema. Las llamadas básicas y las llamadas complementarias. La implementación de extensiones en los archivos. Métodos de acceso.
- 10.3. Los directorios. Su estructura. Las diferencias con los archivos. Tipos de directorios.

Objetivo de la Unidad: Describir qué se entiende archivos, las distintas operaciones que se pueden realizar. Describir los métodos de acceso. Describir que se entiende por directorio de archivos. Describir los diferentes tipos de directorios y asociarlos a los diferentes sistemas operativos.

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Explicar y reconocer los distintos tipos de archivos.
- Conocer las características de cada una de las herramientas descriptas.
- Comprender y explicar las diferencias o similitudes existentes entre los directorios y los archivos que contiene.
- Conocer y explicar la terminología específica empleada en este módulo.

UNIDAD TEMATICA 11: Relocación

- 11.1. El cabezal del programa. El texto, los datos, el texto reubicable, la tabla de símbolos, la tabla de strings
- 11.2. Las direcciones para la reubicación. El índice y el SPARE.
- 11.3. Secciones linkeables y segmentos ejecutables. La relación. Su ubicación en memoria.
- 11.4. El checksum. Su importancia y la forma de calcularlo.

Objetivo de la Unidad: Describir qué se entiende por relocación. Describir las diferentes funciones de las secciones y los segmentos.

Objetivos del Aprendizaje: Después de estudiar éste módulo, el alumno estará en condiciones de:

- Explicar y reconocer los distintas formas de revocación existentes.
- Conocer y explicar porque es necesaria la relocación.
- Conocer y explicar la terminología específica empleada en este módulo.

PROGRAMA ANALÍTICO. CONTENIDOS PRACTICOS:

Listado de trabajos prácticos a realizar

UNIDADES	TRABAJO PRÁCTICO
1, 2, 3, 4	Ampliación de la capacidad de memoria del Sodium
5, 6, 7, 8	Desarrollo de system calls, programación e implementación
9, 10, 11	Desarrollo de un file system.

La máxima cantidad de alumnos para formar un grupo es de hasta cinco alumnos.

Se buscará desestimar los trabajos prácticos unipersonales.

Diariamente los alumnos contarán con tiempo de clase afectado a aclarar dudas que puedan



presentarse en la resolución de los prácticos.

Se requerirá la presentación obligatoria de los trabajos prácticos.

4. BIBLIOGRAFÍA

Título	Autor(es)	Editorial	Año Edición	Ejemplares disponibles en UNLaM
Los microprocesadore INTEL Arquitectura, programación e interfases.	Barry B. Brey	Prentice Hall – Quinta Edición	2010	
Microprocesadores avanzados	José Angulo y Enrique Funke	Paraninfo – Cuarta Edición	2000	
Operating Systems Internals and Design Principles (6th Edition)	Stallings, William	Prentice Hall	2008	
Operating Systems Concepts (8th edition)	Silberschatz, J.L. Galvin P. B and. Gagne	John Wiley & Son. inc	2009	
Sistemas Operativos Principios de diseño (Fifth Edition)	Stallings, William	Prentice Hall	2006	
Real-Time Systems and Programming Languages (Fourth Edition) Ada 2005, Real-Time Java and C/Real-Time POSIX	<u>Alan Burns</u> and <u>Andy Wellings</u>	Addison Wesley Longmain, ISBN: 978-0-321-41745-9	April 2009	
Electrónica digital: aplicaciones y problemas con VHDL	ARTIGAS MAESTRE, José Ignacio...[et al.]	Prentice-Hall, ISBN: 84-205-3222-3	2002.	
Programación Linux 2.0	Rémy Card, Eric Dumas y Frank Mével -	Ediciones Gestión 2000 S.A. – En castellano.	2008	
Rapid system prototyping with FPGAs	Cofer, R. C , Harding, Benjamin F.	Boston : Elsevier/Newnes ISBN: 07502668666,	2005.	
Advanced 80386 Programming Techniques	James L. Turley	Osborne McGraw Hill	2008	
<i>Parallel Computer Architecture: A Hardware/Software Approach.</i>	Culler, David E., Sing, Jaswinder Pal and Gupta, Anoop..	Morgan Kauffmann Publishers. San Francisco, CA.. ISBN 1-55860-343-3	1996	
<i>Computer Architecture: A Quantitative Approach:</i>	<i>Second Edition.</i> Hennesy, John L. and Patterson, David	A. Morgan Kauffmann Publishers. San Francisco, CA. ISBN 1-55860-329-8.	1996.	
Computer Organization and Architecture (Fifth Edition)	Stallings, W	William.. Prentice Hall. New Jersey. ISBN 0-13-081294-3.		
FPGA prototyping by VHDL examples Xilinx Spartan-3 version	Chu, Pong P.,	Hoboken (New Jersey) Wiley-Interscience cop.	2008	
Diseño de procesadores con VHDL	Grediaga Olivo, Angel,	San Vicente del Raspeig Publicaciones de la Universidad de Alicante D.L	2007	
<i>Distributed Systems: Principles and Paradigms, (2th Edition)</i>	Andrew S. Tanenbaum, Maarten Van Steen,	Prentice Hall,	2006.	
Manuales de INTEL – Intel 64 and IA-32 Architectures	Intel Corporation	Intel Corporation	2009	
Distributed Systems. Concepts and Design. Int. Computer Science Series. (Third Edition)	Coulouris, G.F., Dollimore J. y Tim Kindberg	Addison Wesley	2001 (disponible en castellano)	



UNIVERSIDAD NACIONAL DE LA MATANZA

Distributed Operating Systems, Concepts and design. (Third Edition.)	R. Chow, T. Johnson;	Addison-Wesley..	2001	
The Grid: Blueprint for a New Computing Infrastructure (2 nd Edition)	Ian Foster and Carl Kesselman	Morgan - Kaufmann	2000	
MPI- The complete Reference (2 nd Edition) Vol: 1 y 2)	Marc Snir, Steve Otto, S. Huss-Lederman David Walker y J. Dongarra	The MIT Press	1998	