

# **Construcción de un sistema operativo didáctico**

Hugo Ryckeboer  
[h\\_ryckeboer@yahoo.com.ar](mailto:h_ryckeboer@yahoo.com.ar)

Nicanor Casas  
[ncasas@unlam.edu.ar](mailto:ncasas@unlam.edu.ar)

Lic. Graciela De Luca  
[gdeluca@unlam.edu.ar](mailto:gdeluca@unlam.edu.ar)

**Universidad Nacional de la Matanza**

**Departamento de Ingeniería e Investigaciones Tecnológicas**

**Dirección: Florencio Varela 1703 - Código Postal: 1754 - Teléfono: 4480-8900/8835**

## **Resumen**

El propósito principal de la construcción de un sistema operativo de características didácticas, es la de permitir que los alumnos de materias afines puedan comparar el comportamiento de diferentes algoritmos, en primera instancia, de administración de procesos, utilizando para ello una estructura de parametrizaciones en tiempo de ejecución. Esto les permitirá realizar pruebas del comportamiento de los diferentes algoritmos en el manejo de procesos pudiendo realizar análisis de los puntos de conflicto, tan difíciles de experimentar a través de desarrollos manuales

La estructura para la resolución de conflictos se basa en la jerarquía excepción, interrupción y llamadas al sistema especificando en que momentos un sistema operativo puede modificar la misma para obtener una mejor performance de acuerdo a nivel de aceptación que se espera del mismo por parte de los diferentes usuarios

**Palabras Claves:** Núcleo, Cargador, Unix, FIFO, SJF, Round Robin, Prioridades, FAT, IVT, BDA, DPL, GDT, PIC, TSS, RTC, IDT

## **1.- Introducción**

La pregunta fundamental que nos hemos planteado es porque el sistema operativo, en el que estamos trabajando, es de características didácticas y hemos encontrado una serie de puntos que apoyan esta idea:

- a) porque nos permite apoyar a la materia específica, en nuestro caso Sistemas Operativos, a través de la visualización de las tablas del sistema; la entrega de información sobre los eventos que ocurren dentro del sistema y la posibilidad de cambiar la configuración sin detener la ejecución permitiendo dar soluciones a los numerosos problemas que se presentan y que solo podemos resolverlos a través de ejercicios manuales.
- b) porque está desarrollado por los alumnos, y esto implica conocimiento de la particularidad en la que se está trabajando y el entorno que la contiene.
- c) porque se realiza una transferencia de conocimientos, a través de interfaces para cada funcionalidad, lo que permite que el alumno conciba y escriba módulos alternativos.
- d) porque los alumnos que intervinieron en la evolución del sistema operativo proveen soporte a los nuevos desarrolladores, a través de sus documentos, y además muchos de ellos informan y mantienen relación con los que continúan con lo que ellos comenzaron.

Existe una marcada diferencia entre los sistemas operativos de uso general (llámense Linux y Windows) con respecto a los libros más afamados de la materia Sistemas Operativos. Esta diferencia corresponde a las funciones genéricas desarrolladas por sus creadores y las necesidades de diseño lo que lleva a que se dificulte la comprensión de los alumnos sobre el funcionamiento de un sistema operativo básico. El SODIUM pretende minimizar esa diferencia permitiendo realizar las mismas funciones apelando a los algoritmos básicos.

## **2.- Contexto**

El sistema operativo SODIUM nace como un proyecto de cátedra, que tenía como objetivo el desarrollar un sistema operativo básico que permitiera a los alumnos realizar desarrollos en lenguaje C y Assembler.

Los análisis realizados sobre los sistemas operativos vigentes de mayor uso en el ámbito comercial nos impulsaron a dejar de lado al sistema operativo Linux porque la estructura del mismo nos imposibilitaba el hecho de parametrizarlo de tal manera que nos permitiera pasar de un algoritmo a otro y principalmente no nos permitía generar experiencia sobre diferentes arranques y el comportamiento de las diferentes tablas que se generan en el arranque del sistema.

El sistema operativo Windows no nos permitía ingresar a su estructura básica para modificarla, al momento de comenzar con el proyecto, por lo que se excluía de la posibilidad de tomarlo como parámetro.

Se analizaron otros sistemas operativos de características didácticas como ser Minix, Nachos, Zeus Os, Minirighi pero ninguno de los mismos parametriza en tiempo de ejecución para evitar las compilaciones del sistema.

A instancia de los alumnos, el proyecto cobra vida y se decide realizar un sistema operativo propio comenzando desde cero y este sistema operativo recibe el nombre de SODIUM

## **3.- Objetivos**

Tiene como objetivo principal el de permitir comparar diferentes algoritmos de administración de procesos, y en futuras etapas, de administración de memoria, de manejo de dispositivos de entrada/salida y diferentes sistemas de archivos, utilizando para ello un sistema de parametrizaciones en tiempo de ejecución.

El segundo objetivo del SODIUM es el de trabajar los algoritmos en su forma original, tal como fueron planteados por sus creadores, y siguiendo la estructura básica, de esta forma evitando las adaptaciones características de los sistemas operativos más conocidos, que compiten por la penetración en el mercado informático.

El tercer objetivo es posibilitar a su vez que los diferentes elementos constitutivos de la computadora puedan ser virtualizados a fin de que los algoritmos puedan funcionar y ser consultados en su forma más pura sin tener que depender de otros elementos que puedan deteriorar su rendimiento.

## **4. - Metodología empleada**

La metodología empleada para el desarrollo del sistema operativo SODIUM consiste en tres funciones bien diferenciadas.

- a) El grupo de investigación realiza el análisis, el estudio de los elementos básicos que deben ser incorporados al sistema operativo.
- b) Los alumnos realizan la programación e investigación de las características funcionales de los módulos a construir y generan los lotes de prueba necesarios para el testing.

- c) Por último el equipo de investigación analiza y selecciona los mejores trabajos realizando los ajustes necesarios para la estandarización del sistema operativo y procede a la instalación de los mismos preparando la estructura básica para el nuevo ciclo lectivo.

## **5.- Estructura del Kernel**

Después de un largo proceso de evaluación sobre cual sería la estructura del sistema operativo para estudio, la estructura monolítica o la estructura por capas o niveles o jerárquica, se decidió que el sistema operativo comenzaría por la estructura monolítica porque la misma presenta las siguientes ventajas:

- a) es la que más rápidamente permite una toma de conocimiento y de experiencia que faltaba en el equipo de trabajo y que la estructura por capas o niveles era más compleja de transmitir a aquellos que están incursionando por primera vez en la construcción de un sistema operativo.
- b) Es la estructura de los primeros sistemas operativos constituidos fundamentalmente por un solo programa compuesto de un conjunto de rutinas entrelazadas de tal forma que cada una puede llamar a cualquier otra, es decir que tienen amplia libertad para comunicarse entre ellas.
- c) Es la estructura que mejor conocen los alumnos porque en las materias que anteceden a ésta es la más vista. La utilización de un procedimiento se llama directamente, no necesita mensajes, por lo que es más rápido y más fácil de conceptualizar.

Es sabido que todas las actualizaciones que se realicen en el futuro implican recompilar todo el núcleo, pero eso permite a los desarrolladores entender mejor la estructura del sistema manteniendo la configuración del Makefile.

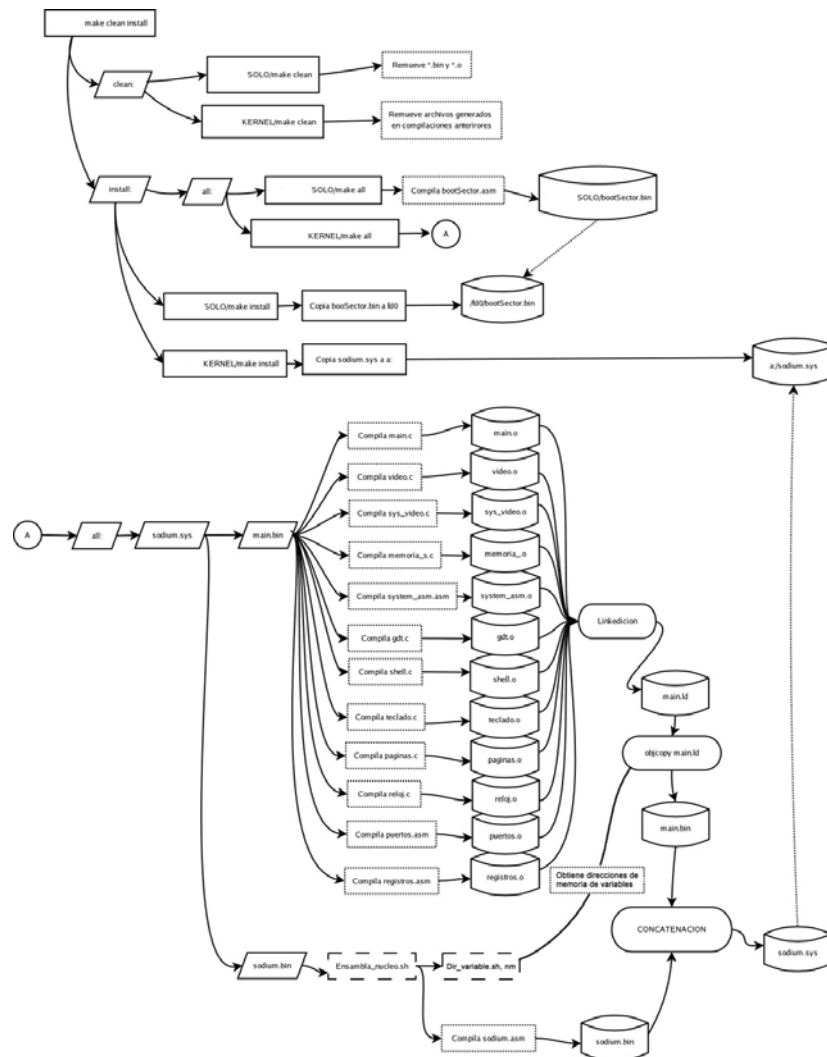
## **6.- El sistema operativo y los anillos de seguridad**

Si bien consideramos que el uso de todos los anillos de protección que provee la arquitectura INTEL es de gran interés didáctico, actualmente, y con el objetivo de lograr avances en otras funcionalidades básicas, todas nuestras tareas se encuentran ejecutando a nivel 0 dada la estructura monolítica adoptada, sin embargo el proyecto contempla que al final del primer año tener estructurada la plataforma de ejecución en dos niveles (0 y 3), ejecutando el kernel en el nivel 0 y los procesos usuario en el nivel 3.

Posteriormente se podrá trabajar con cualquiera de los anillos permitiendo la parametrización correspondiente de los mismos y pudiendo simular el comportamiento del sistema operativo Linux.

## **7.- Diagrama de compilación, enlazado, y armado del disquete de pruebas.**

En la página siguiente se muestra la forma en que se compila, se producen los enlaces correspondientes y se arma el diskette de prueba, con la estructura básica del Make.



## 8. Conclusiones

De acuerdo a lo expuesto el sistema operativo SODIUM se convierte en una herramienta útil para los alumnos y profesores de las materias relacionadas. De él pueden obtenerse excelentes comparaciones para el análisis de los algoritmos más importantes. Hasta el momento sólo se relaciona con la administración de procesos, pero todo el equipo de investigación se encuentra abocado a generar la posibilidad de comparar algoritmos de administración de memoria y de administración de entrada salida generando los consecuentes drivers, manteniendo la característica de la parametrización.

## 9. Bibliografía

- [ANG98] Angulo José M. y Funke Enrique – Microprocesadores avanzados 386 y 486 – Introducción al Pentium y Pentium – Pro Editorial Paraninfo – Cuarta Edición
- [BRE00] BRE00- Brey Barry B. – Los Microprocesadores Intel – Editorial Prentice Hall – Quinta Edición.
- [CAR01] Card Rémy, Dumas Eric, Mével Franck - Programación Linux 2.0 API del sistema y funcionamiento del núcleo – Enrolles y Ediciones Gestión 2000 S.A.
- [DEI90] Deitel Harvey M. – Introducción a los Sistemas Operativos - Addison-Wesley Iberoamericana – Segunda Edición

- [INTEL] Manual de microprocesadores 386 y 486 y Pentium.
- [MIL94] Milenkovic Milan – Sistemas Operativos Conceptos y diseño – Mc Graw Hill – Segunda edición
- [SIL97] Silverschatz, Avi; Galvin, Peter – Operating System Concepts – Addison-Wesley Longman – Fifth Edition
- [SMC00] Standard Microsystems Corporation – Application note 6.12
- [STA98] Stallings Willams – Operating Systems Internals and design principles – Prentice Hall International – Third Edition
- [TAN97] Tanenbaum Andrew S., Woodhull Albert S. – Operating Systems Design and Implementation – Prentice Hall – Second Edition
- [TAN03] Tanenbaum Andrew S.– Sistemas Operativos Modernos – Pearson Education – Segunda Edición
- [TUR03] Turley James L. – Advanced 80386 programming techniques – Osborne McGraw Hill

### Internet

- [01] Página de la cátedra [www.souniver.com.ar](http://www.souniver.com.ar)
- [02] INTEL 8272 Floppy disk Controller  
<http://andercheran.aiind.upv.es/~amstrad/docs/i8272/8272sp.htm>
- [03] BONAFIDE os development (detecting floppy drives)  
[http://www.osdever.net/tutorials/detecting\\_floppy\\_drives.php?the\\_id=58](http://www.osdever.net/tutorials/detecting_floppy_drives.php?the_id=58)
- [04] BONAFIDE os development (howto program the DMA)  
[http://www.osdever.net/tutorials/howto\\_dma.php?the\\_id=63](http://www.osdever.net/tutorials/howto_dma.php?the_id=63)
- [05] The Unix File System <http://www.isu.edu/departments/comcom/unix/workshop/unixindex.html>
- [06] Cátedra de la profesora Gloria Guadalupe González Flores de la Universidad Juárez Autónoma de Tabasco trabajo sobre DMA de Roberto García García  
[http://mx.geocities.com/antrahxg/documentos/org\\_comp/procesador.html#inicio](http://mx.geocities.com/antrahxg/documentos/org_comp/procesador.html#inicio)
- [07] Tecnología del PC- La Placa base <http://www.zator.com/Hardware/H2.htm>