

# VisualOS

## Resumen

El propósito de este proyecto es desarrollar una herramienta gráfica que permita el estudio y comprensión del funcionamiento real de un sistema operativo moderno.

El programa que se presenta permite observar los aspectos más relevantes de un sistema operativo en funcionamiento. También permite ver cómo funcionan y cómo interactúan los tres sistemas mas importantes: planificación de procesos, gestión de memoria y Entrada/Salida. Las representaciones son dinámicas pudiendo capturarse algunas de las gráficas, para su posterior estudio, mostrar el funcionamiento del sistema en "vivo"o experimentar directamente con él.

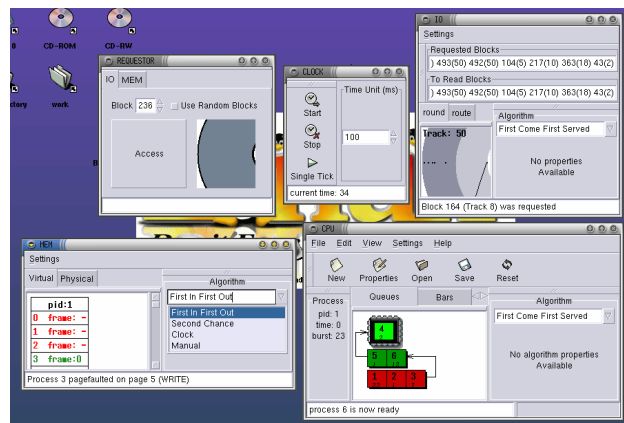
Un sistema operativo consta de muchas partes íntimamente relacionadas, de manera que su buen funcionamiento depende tanto del correcto funcionamiento de cada una de ellas como de su correcta interacción. Este programa permite tanto su estudio global como el estudio concreto de alguna de sus partes.

Tradicionalmente cada parte de un sistema operativo se estudia por separado, siendo muy difícil ofrecer una visión global e integrada de todas ellas, así como de los problemas de la concurrencia, debido a la difícil representación del sistema en funcionamiento.

Hasta ahora la única manera de comprender realmente estos conceptos era escribiendo y modificando el código de un sistema operativo real para luego analizar su funcionamiento. Con este propósito se escribieron sistemas operativos como MINIX, NACHOS o TUNIX. El problema que tiene este método es que requiere mucho tiempo, y para ser realmente útil se necesita un gran dominio del lenguaje de programación utilizado, que permita centrarse en la lógica del problema. Esta nueva herramienta permitirá la realización de laboratorios en los cuales se podrá, en poco tiempo, comprender conceptos fundamentales del funcionamiento real de los sistemas operativos.

En el ámbito docente, el presente programa convierte en más sencillas las tareas

anteriores. Facilita al profesor el trabajo de realizar gráficas y otras representaciones que expliquen los conceptos importantes del sistema operativo, y permite que el alumno los vea más claramente, porque podrá realmente ver, por ejemplo, cómo los distintos procesos se disputan el procesador y la memoria a la vez que pretenden ser los primeros en leer del disco y cómo es el sistema el que gestiona estos recursos. Todo lo verán en conjunto y no fragmentado como se puede explicar en una clase tradicional. Por supuesto también se pueden desactivar algunas de las partes para poder ver más claramente el funcionamiento de las demás. Incluso se pueden mostrar los distintos subsistemas en máquinas diferentes para realizar prácticas en las que distintos alumnos se hagan cargo de cada uno, eligiendo el algoritmo que quiere utilizar o incluso tomando las decisiones manualmente.



Por otro lado, también se podrá utilizar como banco de pruebas, haciendo más fácil y flexible el método de modificar y escribir código, ya que la aplicación está escrita de forma modular y tomando especial interés en hacer lo mas sencillo posible la adición de nuevos algoritmos o modificación de los ya existentes. Para poder hacer esto no será necesaria la comprensión del resto del código, solamente será necesario aprender el interfaz que los algoritmos en cuestión utilizan, una estructura de datos y algunas funciones. Incluso se han escrito funciones para ocultar al implementador de los algoritmos la utilización de GTK+ como librería de elementos gráficos.

Para hacer lo más fácil posible el mantenimiento de la documentación interna y

asegurar así la calidad de esta, se ha utilizado un sistema que genera código DocBook (SGML) a partir de comentarios estructurados en el código fuente del programa denominado "gtk-doc". DocBook, es un estándar en la industria (SUN Microsystems, Hewlett Packard, O'Reilly ...), y por su carácter semántico permite escribir los documentos una sola vez y de forma automática obtenerlos en diferentes formatos: HTML, PostScript, PDF, DVI, Braille ...

El sistema de control de versiones utilizado ha sido CVS (Concurrent Version System), que es el estándar *de facto* en el ámbito del software libre. El CVS permite gestionar los cambios de colecciones completas de archivos.

Gracias a las herramientas "automake" y "autoconf" la compilación del programa es muy sencilla, ya que se detectan las peculiaridades del entorno de forma automática y sin intervención del usuario.

Se ha considerado importante la utilización de estándares abiertos y herramientas libremente disponibles para facilitar la colaboración en el futuro por parte de los propios estudiantes o de cualquier otra persona que lo desee a lo largo de todo el mundo (mediante Internet). Para lograr estos objetivos se han utilizado herramientas como DocBook (SGML) como formato de documentación, el API de programación GNOME/GTK+, el compilador GCC (GNU C COMPILER) y el sistema de control de versiones CVS (Concurrent Version System).

