

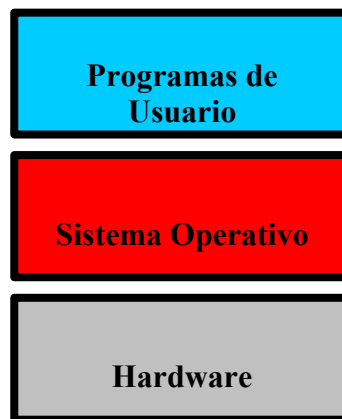
## **1.- INTRODUCCIÓN**

Actualmente, podemos disponer en el mercado desde pequeños Microcontroladores de 4 bits, con una complejidad y prestaciones bastante reducidas que permiten que se puedan programar en lenguaje ensamblador sin demasiado esfuerzo haciendo nosotros todo el trabajo de planificación, manejo de interrupciones, etc. , hasta complejos y caros sistemas de 64 bits en los que programar directamente sobre ellos, sin un Sistema Operativo por debajo que nos facilite las cosas, sería una tarea ardua y totalmente improductiva. Es por eso que debido al aumento de las prestaciones y complejidad en los Microcontroladores, cada vez se hace más necesario abstraerse del Hardware subyacente, permitiendo así que el programador de una determinada aplicación “olvide” la arquitectura sobre la que trabaja, y se pueda ocupar únicamente del Software haciendo uso de los servicios ofrecidos por el Sistema Operativo cuando de utilizar el Hardware se trate, y obviamente, hacer esta labor programando en un lenguaje de alto nivel haciendo uso de API's ya implementadas y todo con una interfaz agradable y completa .

## **2.- ¿QUÉ SE LE DEBE PEDIR AL SISTEMA OPERATIVO?**

Como se ha dicho anteriormente, el hecho de utilizar un Sistema Operativo permite que el tiempo empleado para el diseño y creación de un determinado sistema empotrado disminuya drásticamente, al confiar prácticamente todo el control del Hardware y Software de bajo nivel a él. Así pues un Sistema Operativo tiene que ser un administrador de recursos y una interfaz con el Hardware y el usuario. Debe compartir el Hardware con los distintos procesos, facilitar la E/S, recuperarse de errores, etc, librándonos así de tareas como planificar el sistema, gestionar sistemas de archivos y la memoria, gestionar las interrupciones, dialogar directamente con las controladoras, protocolos de comunicación, etc, haciendo todo esto de una manera autónoma y completamente transparente al programador de aplicaciones.

Además todo esto nos permite hacer programas más portables, con lo que si en el día de mañana se cambia de Microcontrolador, el programa pueda funcionar prácticamente sin problemas sobre otra plataforma y no tener que ser reescrito en su totalidad, bastaría con recompilarlo adecuándolo a la nueva plataforma y si acaso modificar la parte más dependiente del Hardware de nuestra aplicación concreta. Todo esto ha podido ocurrir gracias a que como se ha comentado anteriormente las prestaciones de los Microcontroladores actuales permiten tener potentes Sistemas Operativos, ejecutando varias tareas de un modo “concurrente” que consumen unos recursos no despreciables sin repercutir de una manera excesivamente negativa en el rendimiento global del sistema.



Las ventajas de utilizar un Sistema Operativo que ejecute él nuestras aplicaciones es claro: en primer lugar el programador se puede olvidar de si está programando para un Microcontrolador de Motorola o Hitachi, será el Sistema Operativo el que se ocupará de todo el Hardware, con lo cual se puede centrar únicamente en programar y no en tener que aprenderse el Hardware de cada Microcontrolador cada vez que va a programar, se evita el tener que gestionar las interrupciones, o planificar el sistema (lo cual es crítico si además se trata de Sistemas en Tiempo Real críticos o acríticos), así como la gestión de los procesos y su sincronización, etc.

La gestión del sistema de archivos es transparente (el programador espera que cuando haga un `open()` se abra un fichero independientemente de si está utilizando un sistema de ficheros ext2, o FAT16 o de si el lugar donde esta abriendo el fichero es una memoria o disco de unas determinadas características u otras, así como la gestión de la memoria (el programador no sabrá, y realmente le dará igual, si la memoria esta segmentada, paginada, etc.), evitará implementar él mismo pilas de protocolos (simplemente utilizará Sockets si estos están soportados), o crear funciones para hacer uso de un dispositivo Bluetooth, USB, etc.

### **3.- Un Sistema Operativo para Microcontroladores: *uCLinux***

En un mundo donde cada Euro que se pueda ahorrar es de vital importancia para la competitividad en un determinado sector, una solución idónea es usar Software Libre que permite modificar su código para adecuarlo a nuestras necesidades particulares y hacerlo algo más “nuestro”, así como mejorarlo y que la comunidad de usuarios pueda aprovechar nuestras propias modificaciones. Además en muchos casos, este Software es gratuito lo que permite ahorrarnos el pagar por caras licencias propietarias.

Un ejemplo de esto es *uCLinux* que nació como un derivado del núcleo de Linux 2.0, pensado para Microcontroladores sin MMU. Sin embargo este proyecto ha crecido en reconocimiento y cobertura de arquitecturas de procesador. Actualmente *uCLinux* es un Sistema Operativo que permite incluir entre los núcleos de Linux 2.0, 2.4 y 2.6 así como una colección de aplicaciones de usuario, librerías, etc. Permite multitarea, si bien en muchas aplicaciones esto no es necesario, soporta la pila TCP/IP (y otros muchos protocolos de comunicaciones), sistemas de archivos NFS, ext2, FAT16/32, y todo esto, en algunos casos, en menos de 512 KB de imagen del núcleo gracias al alto grado de optimización del código del kernel y todo de una manera gratuita. Así pues *uCLinux* es una más que interesante opción cuando queramos implementar un Sistema Operativo de contrastada estabilidad y fiabilidad por debajo de nuestras aplicaciones de usuario destinadas a un Microcontrolador.

#### **4.- Bibliografía**

uCLinux Project: <http://www.uclinux.org>

Sistemas Operativos en microcontroladores, **D. Óscar González Martínez:**  
[http://www.depeca.uah.es/alcabot/seminario2006/SEM06\\_SOenMicrocontroladores.pdf](http://www.depeca.uah.es/alcabot/seminario2006/SEM06_SOenMicrocontroladores.pdf)

Introducción a los Sistemas Operativos, Estructura Interna del Sistema Operativo, El Núcleo del Sistema Operativo, Sistemas de Tiempo Real  
(Algunos temas de la Asignatura Arquitectura de Computadores de 4º de Ingeniería de Telecomunicación en la Universidad de Alcalá de Henares),  
**Sebastián Sánchez Prieto, Oscar García Población, Ignacio García Tejedor**