

**Grupo 2 : Alberto Llamas González, Arturo Molina López-Cepero, Carlos Pérez Cruz, David Sánchez Pérez**

## **Ejercicio 3: Reflexión sobre los estados de bloqueo en Linux**

Los procesos *interrumpibles* se encuentran en un estado bloqueado en el que el proceso está esperando un evento como la finalización de una operación de E/S, si un recurso está disponible, o la señal de otro proceso. Los procesos que no realizan ninguna tarea, deben estar en este estado. Además, un proceso sólo puede ser interrumpible si está en modo usuario, puesto que eliminar un proceso en modo kernel podría dar lugar a corromper las estructuras del kernel usadas por el resto de procesos.

Los procesos *no interrumpibles* o *ininterrumpibles*, por contra, están bloqueados y esperando directamente sobre un estado del hardware por lo que no maneja otra señal. Además, normalmente cuando un proceso entra en este estado, significa que está esperando un evento que no tardará en producirse.

Estas son las dos formas que tiene el kernel de Linux de bloquear procesos. Sin embargo, hay veces que conviene poner el estado de un proceso de una forma u otra. Puede ser que necesitemos que un proceso no se ejecute durante cierto tiempo, o que necesitemos que se ejecute hasta que no lo llamemos específicamente. Por ello, esta diferenciación entre estados que hace el sistema es muy útil. Sin embargo, la manera más común de poder bloquear procesos es ponerlo en estado interrumpible excepto en situaciones en las que no podamos controlar las señales como en los dispositivos de E/S.

*¿Cuál es la utilidad de esta diferenciación entre estados bloqueados?*  
Pensamos que la principal utilidad puede ser el ahorro de memoria. De esta forma, podemos bloquear estados que no vayan a ser llamados en un corto plazo de tiempo, ahorrando recursos del sistema dando lugar a un mayor rendimiento. Es decir, tenemos diferentes formas de llamar a procesos y ponerlos de nuevo en la cola de ejecutables. Además, creemos que es una buena solución para aprovechar el tiempo de ejecución de diferentes procesos, en el que la máquina utiliza la lógica para elegir el estado que mejor optimice el proceso.



***Grupo 2 : Alberto Llamas González, Arturo Molina López-Cepero, Carlos Pérez Cruz, David Sánchez Pérez***

**Bibliografía:**

**Sitios web:**

- [http://ocw.innova.uned.es.](http://ocw.innova.uned.es/) (19 de Marzo de 2007). Obtenido de [http://ocw.innova.uned.es/ocwuniversia/Ing\\_tecnico\\_infor\\_sistemas/SO\\_II/contenidos\\_html/ims\\_import\\_3/tema\\_4\\_08.htm](http://ocw.innova.uned.es/ocwuniversia/Ing_tecnico_infor_sistemas/SO_II/contenidos_html/ims_import_3/tema_4_08.htm)
- Programs, I. T.-P. (s.f.). <https://people.cs.uchicago.edu/>. Obtenido de [https://people.cs.uchicago.edu/](https://people.cs.uchicago.edu/~shanlu/paper/sosp15-itask.pdf): <https://people.cs.uchicago.edu/~shanlu/paper/sosp15-itask.pdf>
- Sovani, K. (28 de Julio de 2005). LinuxJournal. Obtenido de linuxjournal.com: <https://www.linuxjournal.com/article/8144>

**Libros:**

- Stallings, W. (2005). Sistemas Operativos. Madrid: Pearson Educacion, S.A.