

Simulador de un kernel

En la parte teórica de la asignatura de Sistemas Operativos se estudian cuáles son los principales elementos de un kernel: gestión y planificación de procesos, sistema de memoria y entrada/salida. Además, en el último tema se expondrán las medidas de seguridad que se implementan en los kernel y el tipo de ataques que han aparecido en los últimos años.

El objetivo de esta práctica es desarrollar un simulador de un kernel. El trabajo a realizar se divide en 5 partes, 3 obligatorias y dos opcionales. Asimismo, con esta práctica se trabajará el lenguaje de programación C, que también se utiliza en otras asignaturas de esta especialidad.

1ª parte: Arquitectura del Sistema (2 puntos). Obligatoria

En esta primera parte se definirá la arquitectura del sistema. Sobre todo aquí hay que preparar el sistema para realizar las siguientes partes de la práctica. Para ello es necesario definir las estructuras de datos necesarias para realizar las simulaciones e implementar el *thread* principal que gestionará todo el sistema. Este *thread* tendrá varias tareas, por ejemplo, gestionará el reloj del sistema.

2ª parte: Planificador (3 puntos). Obligatorio

El planificador (o *scheduler*) gestiona la ejecución de los procesos. En la construcción de este subsistema se pueden utilizar diversas políticas y arquitecturas. En esta parte, teniendo en cuenta la descripción del hardware entregado, se deberá implementar una (o varias) política que maximice la eficiencia y el rendimiento de los procesos.

3ª parte: Gestor de Memoria (3 puntos). Obligatorio

Un kernel utiliza direcciones virtuales y físicas para gestionar la memoria. Los procesos utilizan direcciones virtuales y el procesador realiza la traducción para obtener las direcciones físicas correspondientes a dichas direcciones utilizando el TLB y la MMU. En esta parte se deberá implementar el proceso descrito. Para ello, cada proceso que se ejecute tendrá un listado de accesos a la memoria y el sistema simulado deberá realizar un proceso de traducción para realizar lecturas o escrituras en la memoria física.

4ª parte: Controlador de un disco flash (1 punto). Optativo

En esta parte se implementará la capa para acceder a dispositivos de tipo Flash, denominada *Flash Translation Layer* (TFL). El objetivo de esta capa es evitar el denominado efecto *wearing*. Al igual que en la parte anterior, los procesos utilizan direcciones lógicas para acceder al disco, que se traducen en direcciones físicas utilizando el TFL.

5ª parte: Seguridad (1 punto). Optativo

En teoría se expondrán varios métodos para garantizar la seguridad de los kernels. En esta parte se seleccionará uno o varios de ellos y se implementará en el simulador. El simulador deberá evitar un ataque del profesor que garantice la seguridad.

Documentación

Coincidiendo con las partes 2 y 4 se entregará un breve informe en el que se detallará el diseño y la implementación de cada parte y de la anterior.

Notas

Criterios a tener en cuenta para la realización de la práctica:

- Para realizar esta práctica utilizaremos ANSI C.
- Se recomienda el uso de un sistema operativo de tipo UNIX, pero no es obligatorio.
- Podrás utilizar los ordenadores que hay en el laboratorio, pero se recomienda utilizar tu propio ordenador.
- Necesitarás un repositorio git. Se recomienda utilizar una cuenta de Gitlab pero se puede utilizar otra.
- Existe la posibilidad de que un alumno pueda proponer una práctica diferente a la planteada.