**SISTEMAS OPERATIVOS**

**2º Curso – Dobles Grados**

**Ejercicios del Tema 1**

**Leandro Jorge Fernández Vega**

**1. Uno de los mecanismos de virtualización ligera que suministra el kernel de Linux es el de namespaces. Indicar cuál es la función básica del citado mecanismo.**

Los *namespaces* son módulos que permiten aislar determinados procesos para que sean únicamente vistos por otros procesos del mismo *namespace*. Es decir, ofrecen vistas locales a procesos.

Cada proceso tendrá un PID interpretable de distinta manera dependiendo de dónde se ponga la referencia. Por tanto, los *namespaces* favorecen una construcción modular del sistema, ya que dentro de uno mismo se agrupan procesos relacionados lógicamente.

**2. En relación a la arquitectura monolítica:**

**a) Cuál es la principal razón para que esta arquitectura sea la “arquitectura de ejecución” más utilizada en los sistemas operativos más comúnmente utilizados, y cuál es el principal inconveniente.**

Los sistemas monolíticos siguen un tipo de estructura que consta de un único ejecutable muy extenso que consta de colecciones de procedimientos enlazados entre sí, donde todos los procesos utilizan el mismo espacio de direcciones.

Por tanto, la principal ventaja es la eficiencia, al haber un solo programa en ejecución, pero son sistemas pocos robustos debido a la falta de modularización y la poca consideración hacia las dependencias entre procesos. Además, tienen gran dificultad de comprensión y mantenimiento, especialmente en las relaciones entre procesos.

**b) Las estructuras monolíticas actuales, como ocurre en Linux y Mac OS X, permiten algún mecanismo de adaptación de la funcionalidad del kernel en tiempo de ejecución Indicar qué mecanismo es y en qué consiste. NO ESTÁ BIEN**

PMCTrack es una herramienta de código abierto para Linux que permite la monitorización de diversos aspectos de las aplicaciones en tiempo de ejecución empleando contadores hardware u otros recursos hardware de monitorización, como sensores de temperatura o de consumo energético. Para ofrecer acceso a la información de monitorización al usuario final y al SO, esta herramienta consta de componentes tanto de espacio de usuario como de kernel. Su interfaz, que es independiente de arquitectura, facilita la realización de optimizaciones en el SO, principalmente orientadas a la planificación de procesos y a la gestión de recursos.

**3. Explicar cuáles son las diferencias entre hipervisores nativos (Tipo I) e hipervisores anfitrión (Tipo II). ¿Cómo se hace para que un SO que está diseñado para ejecutarse en modo kernel/supervisor se ejecute en otro nivel de privilegio?**

La diferencia principal es que los hipervisores tipo 1 se implementan directamente sobre el hardware, mientras que los hipervisores tipo 2 se hacen sobre un SO anfitrión que sigue estando en funcionamiento mientras éste se ejecuta.

El hipervisor debe gestionar los niveles de privilegio del procesador. Puede hacerlo asistido por:

* Software:

1. *Traducción binaria:* se pone el SO en un anillo más protegido que el usuario para atrapar las trampas del SO en VMM.
2. *Tablas de páginas sombra:* proyecta la memoria del anfitrión en la memoria de la máquina real, para que este tenga acceso a las tablas de página hardware.
3. *Emulación de E/S:* las solicitudes del sistema emulado se entregan al hardware real.

* Hardware: usa un anillo con privilegio mayor que el 0 y no necesita ni emulación, ni traducción binaria. Ej. *Intel Virtualization Technology*.
* Ambos.

**4. Los kernel actuales de Linux incluyen un mecanismo denominado espacios de nombres (namespaces). Indicar cuál es la función básica del citado mecanismo y en concreto la del PID namespaces.**

Funcionamiento explicado en Ejercicio 1.

Los PID *namespaces* agrupan los ID de los procesos del sistema.

**5. Explicar cuál es el principal inconveniente de una arquitectura monolítica.**

Explicado en Ejercicio 2a).

**6. Los kernel actuales de Linux soportan dos mecanismos básicos para la virtualización:**

**a) Indicar que mecanismos son y qué responsabilidad tienen cada uno desde el punto de vista de la gestión de recursos.**

* *Namespaces:* son módulos que permiten aislar determinados procesos para que sean únicamente vistos por otros procesos del mismo *namespace*. Es decir, ofrecen vistas locales a procesos, agrupando los relacionados lógicamente.
* *Control Groups:* limitan recursos a todos los procesos miembro de un mismo *cgroup*. Es decir, controlan los recursos asignados a una determinada tarea o grupo de tareas.

**b) Cuál es el objetivo de los espacios de nombres de identificadores de procesos (pid namespaces), y cómo se consigue.**

Los PID *namespaces* tienen como objetivo agrupar los identificadores de procesos, de manera que puedan ser utilizados por los procesos que deban utilizarlos y los requieran, y así controlar, además, qué procesos ejecutar y cuáles no. Se crean con las funciones clone() (implementar los hilos: múltiples hilos de control en un programa que se ejecutan concurrentemente en un espacio de memoria compartido) y fork() (genera un duplicado del proceso actual).

**7. Respecto a los tipos de sistemas operativos, indicar:**

**a) Cómo se implementaría un sistema de tiempo compartido.**

Un sistema de tiempo de tiempo compartido debe ser un sistema multiprogramado, ya que consiste en el reparto de la CPU entre varios procesos. Cada proceso tiene asignado un tiempo de ejecución controlado por los tics de un reloj programable. Cuando el contador llega a 0, se le deniega el uso de CPU mediante una rutina de servicio de interrupción de reloj y se llama a scheduler() para reorganizar la actividad CPU.

**b) Qué caracteriza a un sistema operativo SMP (Symmetric Multi-Processing).**

SMP es una característica de SO paralelos que permite a todos los procesadores ejecutar tanto código del SO como de las aplicaciones. Cada procesador tiene su propia ALU, unidades de control y registros, pero comparten memoria, acceso a dispositivos de E/S y reloj.

**8. La mayoría de los sistemas operativos de producción actualmente en uso tienen una estructura de ejecución monolítica, soportan multiprocesamiento simétrico y tienen soporte para tiempo-real. Rellenar la Tabla que se muestra a continuación indicando en cada celda cuáles son las razones de hacerlo así y cuáles son las implicaciones en su construcción.**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Arquitectura**  **de ejecución monolítica** | **Multiprocesamiento**  **simétrico (SMP)** | **Soporte para**  **tiempo-real** |
| **Por qué…** | Mayor eficiencia. | Comunicación y sincronización muy eficiente | Sincronización precisa, respeto de prioridades, posibilita interacción del usuario. |
| **Implicaciones en su**  **construcción ...** | Falta de robustez y modularización, dificultad de comprensión y mantenimiento. | Cada procesador ejecuta una copia idéntica del SO, y tanto aplicaciones como código del SO, por lo que hay que evitar que los procesos con el mismo nombre en distintos procesadores interfieran. | Los procesos deben realizarse dentro del tiempo que se les tiene asignado (deadline) para cumplir su propósito y evitar errores. Esto afecta al planificador, contrucción del kernel, rutinas de interrupción, etc. |