

Sistemas Operativos
Curso 2020-21
Dobles Grados de Informática

Apellidos y nombre: ...David Martínez Díaz

DNI: 44669141J

Observaciones:

- Dispone de 45 minutos para realizar la prueba. Prado esta configurado para cerrar la entrega a las 13:25 h. No se prorrogará el tiempo de entrega.
- Suba a la actividad este documento relleno con sus respuestas. Dicho documento será chequeado con la herramienta antiplagio Turnitin.
- Las cuestiones tienen la misma puntuación dentro de su tema.

Tema 1:

1. En un sistema operativo ciertas funciones son esenciales para el buen funcionamiento del mismo, por ejemplo, mantener siempre memoria disponible para satisfacer las demandas de los procesos y del propio sistema. Explicar la forma en que se consigue realizar dichas funciones en una arquitectura de tipo monolítica.

Para explicar la estructura monolítica, debemos comentar que esta estructura es característica porque todas sus piezas se ejecutan de forma privilegiada (kernel), donde este actúa como una única entidad (programable-ejecutable), por lo que generalmente suele ser complejo la edición y la modificación de esta.

Sobretudo debemos a comentar los hilos kernel de Linux, sustituyendo los demonios característicos de Unix, haciendo que estos SO sean eficientes, fiables y robustas.

Por lo que para llegar a conseguir dicha eficiencia característica de los sistemas monolíticos debemos, tener una estructura compacta que se transmitan los datos rápidamente y con poco coste.

Donde con sus ventajas, se basan en un simple desarrollo de este, fácil de probar, es decir, puedes probarlo de principio a fin con Selenium.

Es fácil de desplegar, por así decirse, es bastante cómodo copiar el conjunto entero entre otras.

2. Respecto al soporte a la virtualización en Linux, indicar cual es el objetivo de poder crear diferentes espacios de nombres de identificadores de los procesos (*PID namespaces*) y cómo se pueden crear.

Los namespaces son utilizados generalmente para poder utilizar referencias o en si, unas especies o tipo de recursos (services) hacia el sistema, estos son característicos porque sueles ser separados en lugares diferentes a la entidad(object) para que desde un namespace

no puede enlazarse o coger los distintos datos o servicios que haya en otro namespace diferente.

Por ello, en cuanto al PID namespaces, esto se utiliza para que cuando utilicemos la virtualización, por ejemplo, Virtual Box, que tiene su propio espacio. Esta es una máquina virtual, que no puede acceder a otra máquina virtual distinta, ya que no puede hacer referencia a sus PIDS.

Además, los PIDS están anidados, esto lo que quiere decirnos es que cuando un proceso se crea, este se le asigna un PID en concreto para cada namespace, desde su primer espacio hasta el inicial con el que empezó.

Tema 2:

3. Sea un sistema Linux sobre un equipo con un único procesador. Tenemos un proceso multihebrado con dos hebras, H1 y H2, ambas con una prioridad de 120. H1 lleva ejecutándose 50 ms y H2 lleva bloqueada 45 ms. Detallar la secuencia de acciones y sucesos que tiene lugar en el momento que se produce el evento por el que estaba esperando H2 y cuál es el estado final del sistema como resultado de la aparición de dicho evento.

Suponemos que, en este caso, habría que esperar hasta el límite de tiempo para H1 en su ejecución para que cuando llegue a su tope, se provoque una transición del H1, que pasaría a estar en un estado bloqueado, ya que H2 estaría haciendo una llamada al sistema, y cuando se produce el evento este pasaría a preparado.

Donde al quedar H1 bloqueado, por lo que se llama a Schedule (), que como la siguiente hebra H2, se encuentra en estado preparado, y con la misma prioridad, es seleccionada, y se produciría un cambio de estado o contexto por el tomaría el control la segunda hebra.

Posteriormente, se ejecutaría H2 con su correspondiente tiempo de ejecución, y posterior a esto, es esperar ya que el H1 está esperando a el evento de H2. Por lo que en algún momento se llegara a desbloquearse el H1 una vez se finalice (se encarga la RSI).

Por último, tras la intervención se reanudaría H1, donde si se puede, cometerá una llamada a un servicio del sistema, que lo bloquee y terminara.

4. - Explicar brevemente 3 mecanismo vistos necesarios para que Linux de soporte a la ejecución de aplicaciones de tiempo-real blandas (soft).

➔ En primer lugar, podemos destacar los Round-Robin, este se caracteriza por ser un planificador que va dejando como una especie de rodajas de tiempo a cada proceso, permitiéndole ejecutarse por un límite de tiempo. Es decir, es como una manera de poder distribuir los procesos, destacando sobretodo, que estas se reparte de manera proporcional y en un orden circular.

➔ FIFO (conocido como First IN y First ON), este planificador se basa principalmente en que los procesos que se ejecuten, seguirán hasta que terminen o hasta que pasen a ser

bloqueados por cualquier motivo, teniendo como idea principal un coste bajo en el “flow”. Estos junto con los Round-Robin, son gestionados por la clase de tiempo-real.

- ➔ Podemos comentar la herencia de prioridades, que ayudaría en este caso, pero también debemos comentar la jerarquía de prioridades, donde van primero las de tiempo real y luego las otras, quedando como últimos los procesos nulos. Además, hablar de la apropiatividad, que estos son como marcas en la ejecución de privilegiados, para poder abarcar un proceso sin utilizar la condición de carrera. Así como la utilización de la planificación oculta.