

Sistemas Operativos
Curso 2020-21
Dobles Grados de Informática

Apellidos y nombre: Daniel Alconchel Vázquez

DNI: 49617109Z

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.

Como sabemos, en una arquitectura monolítica todos los componentes del sistema se ejecutan en modo kernel , es decir, el sistema operativo se escribe como una colección de procedimientos enlazados entre sí en un solo programa binario, y los servicios se obtienen a través del procedimiento ‘protegido’ y usando memoria virtual. Consiste en que los parámetros se colocan en un lugar bien definido (por ejemplo, la pila) y luego se ejecuta una instrucción trap (que se encarga de cambiar la máquina del modo usuario al modo kernel, transfiriendo el control al sistema operativo). Después el sistema operativo obtiene los parámetros y determina cuál es la llamada al sistema que se va a llevar a cabo.

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

Como sabemos, la virtualización nos permite dividir recursos de computación para presentar uno o varios entornos de operación. Para ello, Linux utiliza dos tipos de mecanismos. Uno de ellos son los namespaces, que permiten que se vean propiedades globales del sistema desde diferentes aspectos, es decir, permite aislar en espacios separados al objeto los nombres para referencias de ciertos recursos/servicios del sistema. En concreto, el PID namespaces tiene como objeto en cuestion el PID. Estos se crean con la funciones clone() y fork().

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 ejecutandose 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.

La secuencia comienza con una petición de interrupción por parte de la hebra 2 al procesador. Con esta petición, pide que se ejecute ella misma. El procesador comprueba si el flag TIF_RESCHEDED es igual a 1. En caso afirmativo, llama al planificador, que guardará los datos de la hebra 1 en su correspondiente pcb, debido a que ahora la hebra 2 obtiene una mayor prioridad, debido a que

el consumo de CPU por parte de la hebra 1 es mayor que el de la 2.

A continuación, se cargan los datos del pcb previamente guardado de cuando se encontraba la hebra 2 en ejecución. Aplicando la política apropiativa, la hebra 1 pasará al estado de listo, mientras que la hebra 2 pasará al estado de ejecución.

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

- **Primero en llegar, primero en servirse (FIFO o FCFS):** Consiste en que en el momento en que un proceso pasa al estado de listo se une a la cola de listos. Cuando el proceso que está actualmente en ejecución acaba de ejecutarse, el sistema pasa a ejecutar el que lleve más tiempo esperando en la cola de listos.

- **Turno rotatorio (round robin):** Consiste en que se genera una interrupción de reloj cada cierto intervalo de tiempo. Cuando sucede dicha interrupción, el proceso que se encontraba en ejecución pasa a la cola de listos y se selecciona un nuevo proceso siguiendo el método antes comentado (el FCFS).

Esto permite mejorar el tiempo de espera para ejecutar trabajos cortos, ya que en el modelo anterior podía llevarse bastante tiempo en espera.

- **Herencia de prioridad:** Consiste en que cuando una tarea está bloqueando la ejecución a otra más prioritaria, hereda la prioridad de esta, es decir, cuando un nuevo proceso se añade a la cola y el sistema considera que tiene mayor prioridad, el planificador podría expulsar al proceso en ejecución actual, para pasar a ejecutar el nuevo proceso