

Examen 1º parcial de Sistemas Operativos Avanzados (20/6/2017)

No se permite el uso de documentación. Duración: 40 minutos. F. publicación: 27/6/17. F. revisión: 29/6/17

Nombre y apellidos:

1) Explique de qué manera los mecanismos de Linux *cgroups* y *namespaces* posibilitan la implementación de contenedores.

2) Exprese con la mayor precisión posible tres de las fórmulas asociadas al algoritmo de planificación CFS: **(a)** selección del proceso a ejecutar (**(b)** ¿por qué requiere un RBT?); **(c)** cálculo del tamaño de la rodaja del proceso seleccionado; **(d)** reajuste de un proceso cuando se desbloquea.

3) Explique cómo se gestiona el bit de modificado en una solución de virtualización de la memoria que se base en el esquema denominado “*virtual TLB*” distinguiendo (a) qué acciones se realizan cuando se activa este bit y (b) cuáles cuando se desactiva.

4) Dado un sistema monoprocesador, dibuje la traza de ejecución de los procesos A y B para los siguientes escenarios, indicando si están en modo usuario o modo sistema, si hay cambio de contexto voluntario o involuntario y las diferentes rutinas de eventos que tienen lugar, teniendo en cuenta que:

- La rutina de interrupción de disco (*rutDisc*) tiene asociada una rutina de interrupción software de sistema denominada *rutSSistDisc*.
 - La rutina de interrupción software de proceso de planificación se denomina *rutSProcPlan*.
 - La rutina de interrupción software de proceso de terminación involuntaria se denomina *rutSProcTerm*.
 - La rutina de tratamiento de la llamada a sistema XXX se denomina *rutXXX*.
 - La rutina de tratamiento del fallo de página se denomina *rutFPag*.
 - La rutina de excepción por división por cero se denomina *rutDivCero*.
 - El procesador tiene niveles de interrupción, donde la interrupción de teclado tiene mayor prioridad que la interrupción de disco.
 - Sólo están en marcha los procesos de usuario A y B. La planificación de los procesos se hace por prioridad, y el proceso B es más prioritario que el proceso A.
1. Se trata de un núcleo no expulsivo. El proceso A ejecuta una llamada al sistema *read()*. Dentro de la ejecución de la rutina de dicha llamada al sistema se produce un fallo de página, y dentro del fallo de página hay una división por cero, que provoca que el proceso A tenga que morir. El proceso B está bloqueado. Eventualmente vendrá una interrupción de disco que despertará al proceso B, poniéndolo en ejecución de nuevo.
 2. El mismo caso anterior, pero en un núcleo expulsivo.
 3. Se trata de un núcleo no expulsivo. El proceso B está en ejecución en modo usuario y el proceso A es nuevo. El proceso B realiza una llamada a sistema *wait()* sobre un semáforo que está cerrado, por lo que se quedará bloqueado. Cuando se pone a ejecutar el proceso A, realiza una llamada a sistema *post()*, abriendo el semáforo por el que estaba bloqueado B. Dicho proceso se desbloqueará y, como tiene más prioridad, será planificado.
 4. El mismo caso anterior, pero en un núcleo expulsivo.