Sistemas Operativos 2020-2
Tarea 1. Fecha de entrega: 10 de marzo de 2020

1. Como vimos en clase, para que un dispositivo de E/S haga una operación, el kernel debe indicárselo escribiendo a sus registros de control. Luego, cuando la operación finalizó, el controlador interrumpe al cpu para que tome el control el kernel y así este pueda recuperar (leer) los datos obtenidos; una posibilidad es leyendo los registros de estado del controlador. Esto sería muy lento con dispositivos como el disco duro, ya que las interrupciones serían demasiadas. Menciona la técnica utilizada para solucionar esto.

2. Supongamos que tenemos dos threads del kernel t_1 y t_2. Luego, $t1$ ejecuta **thread_yield** y el calendarizador escoge a t_2 para ejecutarse a continuación. ¿Que thread es el que ejecutará la línea que está justo después de la llamada a **thread_yield** en el código fuente; y porqué?

3. ¿Siempre que se ejecuta una rutina de servicio de interrupción, hay al final de ella un cambio de contexto? Explica tu respuesta.

4. Cuando un proceso espera a un evento, por ejemplo entrada salida/este tiene que dormirse. ¿Quién duerme al proceso?

5. Menciona todos los escenarios en los que podría cambiar el estado de un proceso running a un estado ready.

6. Cuando es atendida una interrupción lo más simple sería que tomara el control del cpu una rutina general que determine que tipo de interrupción y llame a la rutina de servicio de interrupción apropiada. Pero, dado que hay un número predefinido de estas, existe una optimización para manejarlas. Menciona tal optimización.

7. En un semáforo cuyo valor inicial es n. ¿Cuál es el numero máximo de invocaciones del método wait que se pueden completar sobre este?

8. Explica tres formas en las que un proceso de usuario puede pasar parámetros cuando hace una llamada al sistema. ¿Cuál es mas eficiente en cuanto al número y tamaño de los parámetros?, ¿y en cuanto a rendimiento?

9. Hay varias colas en las que puede estar formado un proceso (no solo esperando por ES, puede ser por esperar a la ejecución de un proceso hijo). ¿En cuantas colas puede estar formado un proceso al mismo tiempo y porqué?

10. La comunicación entre procesos por medio de paso de mensajes es mas lenta que por memoria compartida ya que requiere intervención del kernel en cada transferencia de información. Menciona las ventajas y desventajas de la comunicación por memoria compartida contra el modelo de paso de mensajes.

11. Menciona dos instrucciones protegidas (que sólo el código de kernel puede ejecutar), y explica las complicaciones que se derivarían si estas no lo fuesen y los procesos las pudiesen ejecutar directamente.

12. Es posible simular a las instrucciones goto con el modelo de pila. Explica tu respuesta.