

SISTEMAS OPERATIVOS

Segundo Cuatrimestre de 2020

Trabajo Práctico N° 2

1. Problemas: Procesos y Threads

1. En el diagrama de estados de un proceso, ¿dónde interviene el S.O.? ¿Cómo debería ser el diagrama para ver todos los procesos y el S.O?
2. ¿Cuáles son, normalmente, los sucesos que llevan a la creación de un proceso?
3. ¿Cuáles son los pasos que lleva a cabo un sistema operativo para crear un nuevo proceso?
4. En el libro “Operating Systems: Concepts, Policies, and Mechanisms” se definen los siguientes estados para un proceso: ejecutando (Ejecución), activo (Listo), bloqueado y suspendido. Un proceso está bloqueado si está esperando el permiso para usar un recurso y está suspendido si está esperando a que termine una operación sobre un recurso que ya ha conseguido. En muchos sistemas operativos, estos dos estados se unen en el estado de Bloqueado, mientras que el estado de Suspendido representa que el proceso está en memoria secundaria. Compárense las ventajas de ambos conjuntos de definiciones.
5. Describa las diferencias entre planificadores de corto, mediano y largo término.
6. Suponga que cuatro procesos están ejecutando en un sistema con un sólo procesador, las direcciones de inicio de cada uno son las siguientes: 4050, 3200, 5000 y 6700. Las trazas individuales se muestran en el cuadro 1. Realice la traza de entrelazado de ejecución de los procesos. Asuma que el dispatcher es invocado cada 5 instrucciones o por una interrupción y el dispatcher ejecuta 4 instrucciones.
7. La mayor parte de la problemática de la administración del procesador puede describirse con ayuda de un diagrama de transición de estados de los procesos, como el que se expone en la figura 1. El objetivo de este ejercicio es analizar en detalle las relaciones existentes entre los procesos, las rutinas de atención de interrupciones y el planificador de procesos. Debe describir cómo interactúan las rutinas de atención de interrupciones, el planificador de procesos y la base de datos necesaria para cumplir con el diagrama de transiciones. O sea, debe indicar:
 - ¿Cuáles son los eventos que provocan las transiciones 1, 2, 3 y 4.?

Proceso P ₁	Proceso P ₂	Proceso P ₃	Proceso P ₄
4050	3200	5000	6700
4051	3201	5001	6701
4052	3202	5002	6702
4053	3203	5003	<I/O>
4054	3204	5004	
4055	3205	5005	
4056	3206	5006	
4057	<I/O>	5007	
4058		5008	
4059		5009	
4060		5010	

Cuadro 1: Tabla de Procesos

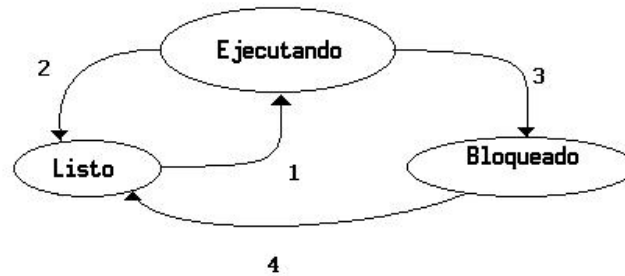


Figura 1: Diagrama de Estado de Procesos

8. Describa las acciones que realiza el kernel cuando cambia el contexto entre procesos.
9. ¿Un proceso puede esperar simultáneamente a más de un suceso ó evento? Justifique su respuesta.
10. Considere el diagrama de estados, que está formado por siete estados (revise el capítulo 3 del libro de Stallings). Suponga que el dispatcher del sistema operativo tiene que elegir un proceso. Hay procesos en el estado LISTO y LISTO/SUSPENDIDO, y que al menos un proceso en el estado LISTO/SUSPENDIDO tiene mayor prioridad que cualquiera de los procesos en el estado LISTO. Dos políticas diferentes se pueden considerar: (1) Siempre se elige un proceso del estado LISTO, para reducir al mínimo el intercambio, y (2) siempre se dará preferencia a los procesos de mayor prioridad. Sugiera una política intermedia que trate de equilibrar la prioridad y el rendimiento.
11. ¿Cómo se pueden comunicar los procesos? Brinde un ejemplo en el cual considere necesario que 2 procesos se comuniquen.
12. Considere la comunicación sincrónica y asincrónica entre procesos. Presente ventajas y desventajas de cada una.

13. Considere un esquema de comunicación entre procesos basado en mailboxes.
 - a) Suponga un proceso P que desea esperar por dos mensajes, uno desde el mailbox A y el otro desde el mailbox B. ¿Cuál secuencia de send y receive debería ejecutar?
 - b) ¿Cuál secuencia de send y receive debería ejecutar P si desea esperar por un mensaje del mailbox A ó del mailbox B (o de ambos)?
14. ¿Cuáles son las características más importantes de los threads? ¿En qué casos los utilizaría? ¿Cuál es la diferencia de los threads a nivel kernel y los que son implementados a nivel usuario?
15. ¿Cuáles recursos son utilizados cuando un thread es creado? ¿y en el caso de un proceso?
16. Proporcione dos ejemplos de programación en los que los mecanismos de multithread no proporcionen un mejor rendimiento que una solución monothread.
17. Una solución multithreaded utilizando threads a nivel de usuario, ¿presenta mejor rendimiento (performance) en un sistema multiprocesador que en un sistema monoprocesador?
18. Considere un sistema multiprocesador y un programa multithreaded utilizando el modelo de threads muchos-a-muchos. Sea la cantidad de threads a nivel usuario mayor que la cantidad de procesadores en el sistema. Discuta sobre el rendimiento en las siguientes situaciones:
 - a) La cantidad de kernel threads alocados al programa es menor que la cantidad de procesadores.
 - b) La cantidad de kernel threads alocados al programa es igual a la cantidad de procesadores.
 - c) La cantidad de kernel threads alocados al programa es superior a la cantidad de procesadores pero menor que la cantidad de threads a nivel de usuario.
19. Un multiprocesador con ocho procesadores tiene 20 unidades de cinta magnéticas. Hay un gran número de jobs (tareas) enviados al sistema donde cada uno requiere un máximo de cuatro unidades de cinta para completar su ejecución. Asuma que cada trabajo se pone en marcha con sólo tres unidades de cinta durante un largo período antes de requerir la cuarta unidad de cinta por un corto período de tiempo hacia el final de su ejecución. Considere que nunca finaliza el envío de tareas de este tipo.
 - a) Supongamos que el planificador del sistema operativo no iniciará un trabajo a menos que tenga las cuatro unidades de cinta disponibles. Cuando un trabajo se inicia, cuatro unidades se asignan de forma inmediata y se no se liberan hasta que finalice. ¿Cuál es el número máximo de trabajos que pueden estar ejecutándose a la vez? ¿Cuáles son la cantidad máxima y mínima de cintas que están ociosas con esta política?

- b) Sugerir una política alternativa para mejorar la utilización de unidades de cinta y al mismo tiempo evitar deadlock. ¿Cuál es el número máximo de tareas que pueden estar ejecutándose a la vez? ¿Cuáles son los límites en el número de unidades de cintas ociosas?