

PRACTICA 02

Código: 20120354I

Apellidos y Nombres: **Moreno Vera Felipe Adrian**

1. En la **figura 2-2** se muestran los estados de tres procesos. En teoría, con tres estados podría haber seis transiciones, dos fuera de cada estado. Sin embargo sólo se muestran cuatro transiciones. ¿Existe alguna circunstancia en la que una o ambas de las transiciones faltantes pudiera ocurrir?

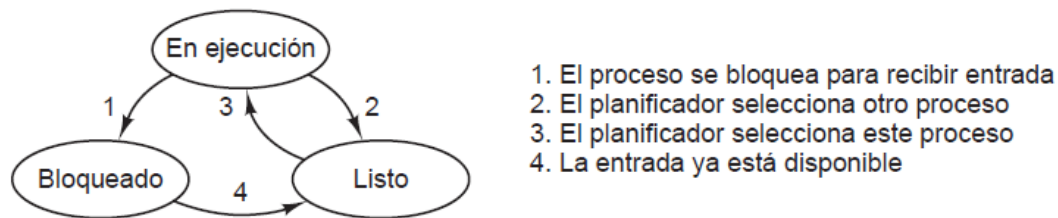


Figura 2-2. Un proceso puede encontrarse en estado “en ejecución”, “bloqueado” o “listo”. Las transiciones entre estos estados son como se muestran.

sol:

La transición de bloqueado en ejecución, es posible. Suponga que un proceso está bloqueado en la Entrada/Salida y el E/S finaliza. Si el CP está inactivo el proceso podría ir directamente de bloqueado a en ejecución. La otra transición faltante de Listo hacia bloqueado es imposible. El proceso Listo no puede hacer E/S o hacia cualquier otra cosa que pudiera bloquearlo. Solo un proceso en ejecución puede bloquearlo.

2. Suponga que debe diseñar una arquitectura de computadora avanzada que hiciera conmutación de procesos en el hardware, en vez de tener interrupciones. ¿Qué información necesitaría la CPU? Describa cómo podría trabajar la conmutación de procesos por hardware.

Sol:

Puede tener un registro que contenga un puntero a la entrada de la tabla de proceso actual. Cuando se completó la E / S, la CPU almacenaría el estado actual de la máquina en la entrada de la tabla de proceso actual. Entonces iría al vector de interrupción para el dispositivo de interrupción y buscaría un puntero a otra entrada de la tabla de proceso (el procedimiento de servicio). Este proceso se pondría en marcha.

3. En todas las computadoras actuales, al menos una parte de los manejadores de interrupciones se escriben en lenguaje ensamblador. ¿Por qué?

Sol:

En general, los lenguajes de alto nivel no permiten el tipo de acceso al hardware de CPU que se requiere. Por ejemplo, se puede requerir un controlador de interrupciones para habilitar e inhabilitar el servicio de interrupción de un dispositivo particular o para manipular datos dentro del área de pila de un proceso. Además, las rutinas de servicio de interrupción deben ejecutarse lo más rápidamente posible.

4. Cuando una interrupción o una llamada al sistema transfiere el control al sistema operativo, por lo general se utiliza un área de la pila del kernel separada de la pila del proceso interrumpido. ¿Por qué?

Sol:

Hay varias razones para usar una pila separada para el kernel. Dos de ellas son las siguientes:

- No se desea que el sistema operativo se bloquee porque un programa de usuario mal escrito no permite suficiente espacio en la pila.
- Si el kernel deja los datos de pila en el espacio de memoria de un programa de usuario, al volver de una llamada de sistema, un usuario malintencionado podría ser capaz de usar estos datos para obtener información sobre otros procesos.

5. Varios trabajos se pueden ejecutar en paralelo y terminar con más rapidez que si se hubieran ejecutado en secuencia. Suponga que dos trabajos, cada uno de los cuales necesita 10 minutos de tiempo de la CPU, inician al mismo tiempo. ¿Cuánto tiempo tardará el último en completarse, si se ejecutan en forma secuencial? ¿Cuánto tiempo si se ejecutan en paralelo? Suponga que hay 50% de espera de E/S.

Sol:

Si cada trabajo tiene 50% de espera de E / S, entonces se necesitarán 20 minutos para completarse en ausencia de competencia. Si se ejecuta de forma secuencial, la segunda terminará 40 minutos después de que comience la primera. Con dos trabajos, la utilización aproximada de la CPU es de 1-0,52. Así cada uno obtiene 0,375 CPU minuto por minuto de tiempo real. Para acumular 10 minutos de tiempo de CPU, un trabajo debe ejecutarse durante $10 / 0.375$ minutos, o aproximadamente 26.67 minutos. Así, ejecutando secuencialmente los trabajos terminan después de 40 minutos, pero corriendo en paralelo terminan después de 26.67 minutos.