1)Responda verdadero o falso y justifique

A. En los ambientes monoprogramados se suele utilizar más frecuentemente las técnicas de prevención de DeadLocks que las técnicas de recuperación ante el mismo.

Falso. En un sistema monoprogramado no puede existir el deadlock. Una de las condiciones para que se produzca deadlock es que haya una espera circular, es decir, dos o más procesos están a la espera de recursos (dibujito acá). Además, la definición de deadlock se enfoca en dos o más procesos. Por lo que en un ambiente monorprogramado no tiene sentido ni prevenirse del deadlock, ni recuperarse, porque nunca sucederá.

B. Los cambios de ejecución de threads ULT de un mismo proceso producen context-switch.

Verdadero. El cambio de hilos de un mismo proceso produce context switches. Esto es porque cada hilo tiene su propio entorno y contexto de ejecución, diferente al de los demás hilos. Sin embargo, este context switch es llamado generalmente thread switch, y su costo es muchísimo menor que el de realizar un process switch. La cantidad de registros que deben cargarse y guardarse es menor, y además, el espacio de direcciones utilizado es el mismo que el proceso padre.

C. Las interrupciones de clock no activan el bit de FLIH por ser internas al procesador.

Falso. Las interrupciones de clock son externas al procesador. Los procesadores tienen un pin (patita) donde se recibe la señal del clock. El clock no es algo interno. Además, las interrupciones de clock son consideradas como las interrupciones de mayor importancia (nivel 0). Estas interrupciones activan el FLIH, que a su vez debe activar la rutina necesaria para atender la interrupción.

D. Es posible que se produzca una interrupción de impresora, una syscall y una interrupción interna al mismo tiempo

Falso en caso de que sea un sistema monoprocesador. Es posible que ocurra una interrupción de hardware externo y una syscall al mismo tiempo. O que ocurran una interrupción interna y una de hardware externo al mismo tiempo. Pero no pueden ocurrir una syscall y una interrupción interna al mismo tiempo.

Verdadero en caso de que sea multiprocesador. En un procesador pueden ocurrir dos interrupciones (hardware externa y syscall), y en el otro puede ocurrir una excepción.

E. Dos procesos pueden producir deadlock por la necesidad de ambos de utilizar dos recursos como la impresora y el procesador.

Falso. El procesador no es un recurso que se vea involucrado en el deadlock. Los procesos no compiten por el procesador, ya que eventualmente tendrán acceso a él.

Si el proceso que está en ejecución necesita un recurso que no está disponible (la impresora) este quedará bloqueado y no se ejecutará hasta conseguir el recurso. En un sistema de multiprogramación, el procesador no puede quedar ocioso sin nada que ejecutar mientras espera. En ese caso, el S.O removerá el proceso bloqueado y pondrá un nuevo proceso de la cola de listos a ejecutarse.

Este nuevo proceso en ejecución puede ser justamente el proceso que desea utilizar el procesador y que tiene la impresora asignada. Por lo que no se produciría un deadlock.

Incluso en un sistema monoprogramado no podría presentarse un deadlock de este estilo.

2) Responda y justifique.

El administrador de un determinado sistema observa que cotidianamente el planificador de mediano plazo ejecuta una cantidad exagerada de tiempo. El mismo ofrece como solución aumentar la memoria central para disminuir los tiempo promedio de ejecución. ¿Cree que esta solución bajará el promedio diario de ejecución del planificado de mediano plazo o simplemente aumentará la velocidad de procesamiento pero el tiempo promedio de ejecución de este planificador seguirá siendo el mismo?

Aumentar la memoria central reduciría la necesidad de uso del planificador de mediano plazo. El planificador de mediano plazo solo es ejecutado cuando hay una gran carga de procesos y, algunos de estos, están bloqueados a la espera de un evento. Estos procesos son swappeados a memoria secundaria para dar paso a otros procesos que sí puedan continuar con su ejecución.

Si hubiera más memoria central, no habría necesidad de enviar los procesos a memoria secundaria, evitándose así la intervención del planificador de mediano plazo.

3) Comparar los siguientes ítems describiendo ventajas y desventajas de desarrollar software sobre cada uno de ellos:

Un sistema operativo de propósito general

De manera directa sobre el hardware.

Sistema operativo de propósito general

|  |  |
| --- | --- |
| Ventajas | Desventajas |
| * Herramientas que hacen el desarrollo de software más fácil para el programador. Generalmente un S.O de propósito general brinda APIs y otros medios para que el programador pueda desarrollar aplicaciones más fácilmente. * No es necesario tener conocimiento exhaustivo del hardware interno. El desarrollador solo tiene que conocer el sistema operativo y sus limitaciones. * Desarrollar una sola vez, correr en varios hardwares distintos. El S.O. sirve como mediador entre los distintos tipos de hardware y el software. | * El software puede consumir más recursos de lo necesario y ser más ineficiente. Al no tener conocimiento del hardware, no se puede exprimir en su totalidad al mismo. * Puede que el sistema operativo intervenga en el funcionamiento del programa, por ejemplo, dando prioridad a otros programas. * Si el sistema operativo de por sí es inestable, el software puede incurrir en problemas. |

De manera directa sobre el hardware

|  |  |
| --- | --- |
| Ventajas | Desventajas |
| * Se pueden aprovechar los recursos más óptimamente al conocer con exhaustividad el hardware. * Menos inestabilidad debido a rutinas mal implementadas, o errores de un intermediario. | * Es más difícil desarrollar software a tan bajo nivel. * El software solo funcionará en dicho hardware específico. * Se deben programar todas las rutinas necesarias para que el software funcione. No hay un intermediario entre el software y el hardware. |