RESPUESTAS A LAS CUESTIONES DE REPASO DEL FINAL DE CADA CAPITULO DEL LIBRO "SISTEMAS OPERATIVOS" DE STALLINGS (4° ED)

Nota: El capítulo 1 (Introducción a los sistemas informáticos) y el capítulo 2 (Introducción a los sistemas operativos) no tienen cuestiones de repaso.

Capítulo 3 - Descripción y control de procesos

Resumen:

La piedra angular de los sistemas operativos modernos es el proceso. La función principal del sistema operativo es crear, administrar y terminar los procesos. Mientras que haya procesos activos el sistema operativo debe velar para que se le asigne a cada uno un tiempo de ejecución en e procesador, por coordinar sus actividades, gestionar los conflictos en las solicitudes y asignar re cursos del sistema a los procesos.

Para llevar a cabo las funciones de gestión de procesos, el sistema operativo mantiene una descripción de cada proceso o imagen de proceso, que incluye el espacio de direcciones en el que se ejecuta el proceso y un bloque de control del proceso. Este último contiene toda la información necesaria para que el sistema operativo administre el proceso, incluyendo su estado actual, los re-cursos que le han sido asignados, la prioridad y otros datos relevantes.

Durante su existencia, un proceso transita por varios estados. Los más importantes son: Listo, Ejecución y Bloqueado. Un proceso Listo es aquél que no está ejecutándose en un momento dado, pero que está preparado para ejecutar tan pronto como el sistema operativo lo expida. Un proceso en Ejecución es aquél que está ejecutándose en el procesador. En un sistema multiprocesador puede haber más de un proceso en este estado. Un proceso Bloqueado es el que está esperando a que termine algún suceso (como una operación de E/S>.

Un proceso en Ejecución puede verse cortado por una interrupción, que es un suceso que se produce fuera del proceso y que es reconocido por el procesador o por ejecutar una llamada del supervisor hacia el sistema operativo. En ambos casos, el procesador lleva a cabo un cambio de modo y pasa el control a una rutina del sistema operativo. Después de que ésta haya terminado su trabajo, el sistema operativo podrá reanudar el proceso interrumpido o cambiar a otro proceso.

Cuestiones de Repaso:

3.1. ¿En qué consiste una traza de instrucciones?

(...) El comportamiento de un proceso individual puede caracterizarse por la lista de la secuencia de instrucciones que se ejecutan en dicho proceso. Dicho listado se llama traza del proceso. (...) (Página 106)

3.2. ¿Cómo son, normalmente, los sucesos que llevan a la creación de un proceso?

Tabla 3.1. Razones para a creación de procesos	
Nuevo trabajo por lotes	El sistema operativo está provisto de un flujo de control de trabajos por lotes, generalmente en cinta o en disco. Cuando el sistema operativo se prepara para coger un nuevo trabajo, leerá la próxima secuencia de órdenes de control de trabajos
Conexión interactiva	Un usuario entra en el sistema desde un terminal.
Creado por el SO para dar un servicio	El sistema operativo puede crear un proceso para llevar a cabo una función de parte de un programa de usuario, sin que el usuario tenga que esperar <por ejemplo, un proceso para control de impresión>.
Generado por un proceso existente	Para modular o para aprovechar el paralelismo, un programa de usuario puede ordenar la creación de una serie de procesos.

(Página 111)

3.3. Describa brevemente cada estado del modelo de procesos de la Figura 3.5.

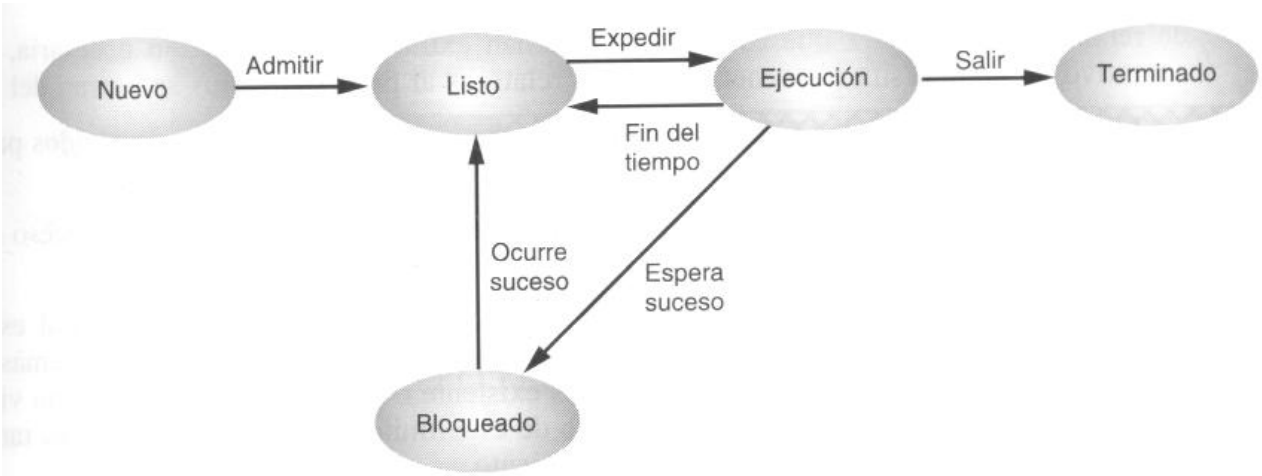

Figura 3.5. Modelo de procesos de cinco estados.

Nuevo: Proceso que se acaba de crear pero aún no ha sido admitido por el sistema operativo en el grupo de procesos ejecutables. Normalmente un proceso nuevo no está cargado en la memoria principal.
Listo: Proceso que está preparado para ejecutarse cuando se de la oportunidad.
Ejecución: Proceso que está actualmente en ejecución. (...)
Bloqueado: proceso que no se puede ejecutar hasta que no se produzca cierto suceso, como la terminación de una operación de E/S
Terminado: un proceso que ha sido excluido por el sistema operativo del grupo de procesos ejecutables, bien porque se detuvo o por alguna otra razón.
(Página 113)

3.4. ¿Qué significa expulsar a un proceso?
Un proceso es expulsado cuando está ejecutando y por alguna razón (por ejemplo: se acabo su cuanto de tiempo) el sistema operativo recupera el uso de procesador y cambia de estado al proceso.

3.5. ¿Qué es el intercambio y cuál es su propósito?
(...) otra solución es el intercambio, lo que significa mover todo o una parte del proceso de la memoria principal al disco (...) (página 117)
Su propósito es poder ejecutar mas procesos que los que entran en la memoria principal.

3.6.¿Por qué la Figura 3.8b tiene dos estados Bloqueado'?
Porque es un modelo que contempla la posibilidad de que existan procesos listos y bloqueados que se encuentren suspendidos en disco.

3.7. Enumere cuatro características de los procesos suspendidos.

a) Un proceso que no está disponible de inmediato para su ejecución.
b) El proceso puede estar esperando o no un suceso. Si lo está la condición de Bloqueado es independiente de la condición de suspensión y el acontecimiento del suceso bloqueante no lo habilita para la ejecución.
c) El proceso fue situado en cl estado suspendido por un agente: bien él mismo, bien el proceso padre o bien el sistema operativo con el fin de impedir si ejecución.
d) EI proceso no puede apartarse de este estado hasta que el agente lo ordene explícitamente. (Página 120)

3.8. ¿Para qué tipos de entidades mantiene el sistema operativo tablas de información destinadas a facilitar la administración?
Memoria, E/S, Archivos y procesos (página 122)

3.9. Enumere tres categorías de información generales en un bloque de control de proceso.

- Identificación del proceso.
- Información del estado del procesador.
- Información del control de proceso. (página 125)

3.10. ¿Por qué son necesarios dos modos (usuario y núcleo)'?
(...) La razón por la cual se utilizan dos modos debe ser clara. Es necesario proteger al sistema operativo y a las tablas importantes del mismo, como los bloques de control de proceso, de las injerencias de los programas de usuario.. En el modo núcleo el software tiene el control completo del procesador y todas sus instrucciones, registros y memoria. Este nivel de control no es necesario, y por seguridad, tampoco conveniente para los programas de usuario. (Página 130)

3.11. ¿Cuáles son los pasos que lleva a cabo un sistema operativo para crear un nuevo proceso?

- Asignar un único identificador al nuevo proceso
- Asignar espacio para el proceso
- Iniciar el bloque de control de proceso.
- Establecer los enlaces apropiados.
- Crear o ampliar estructuras de datos. (Página 131)

3.12. ¿Cuál es la diferencia entre una interrupción y un cepo?
Se pueden distinguir, como hacen muchos sistemas, dos clases de interrupciones del sistema, una conocida simplemente como interrupción y otra conocida como cepo. La primera es originada por algún tipo de suceso que es externo e independiente del proceso que está ejecutándose, como la culminación de una operación de E/S. La segunda tiene que ver con una condición de error o de excepción generada dentro del proceso que está ejecutándose, como un intento ilegal de acceso a un archivo. (Página 132)

3.13. Enumere tres ejemplos de interrupción.

- Interrupción de reloj
- Interrupción de E/S
- Fallo de memoria (página 132)

3.14. ¿Cuál es la diferencia entre cambio de modo y cambio de proceso?
Está claro entonces que el cambio de modo es un concepto distinto del cambio de proceso. Puede producirse un cambio de modo sin cambiar el estado del proceso que está actualmente en estado de Ejecución. En tal caso, salvar el contexto y restaurarlo posteriormente involucra un pequeño coste extra. Sin embargo, si el proceso que estaba ejecutándose tiene que pasar a otro estado (Listo, Bloqueado, etc.), el sistema operativo tiene que llevar a cabo cambios sustanciales en su entorno. (Página 134)

Capítulo 4 -	Hilos, SMP y Micronúcleos

Resumen:

Algunos sistemas operativos hacen una distinción entre los conceptos de proceso e hilo. El primero se refiere a la propiedad de los recursos y el secundo se refiere a la ejecución de programas. Este enfoque puede conducir a una mejora de la eficiencia v hacer más cómoda la programación. En un sistema multihilo. dentro de un único proceso se pueden definir múltiples hilos concurrentes es posible tanto con hilos a nivel de usuario como hilos a nivel de núcleo. El sistema operativo tiene conocimiento de los hilos a nivel de usuario y son creados y gestionados por una biblioteca de hilos que se ejecuta en el espacio de usuario de un proceso. Los hilos a nivel de usuario son muy eficientes. ya que no es necesario un cambio de modo al cambiar de un hilo a otro. Si embargo. en cada instante sólo puede ejecutarse un hilo a nivel de usuario de un mismo proceso, si un hilo se bloquea, se bloquea el proceso completo. Los hilos a nivel de núcleo son hilos internos a un proceso que gestiona el núcleo. Por eso el núcleo tiene conocimiento de ellos, los múltiples hilos de un mismo proceso pueden ejecutarse en paralelo en un multiprocesador y el bloqueo hilo no bloquea todo el proceso. Sin embargo, para cambiar de un hilo a otro es necesario un cambio de modo.

El multiproceso simétrico es un método de organización de un sistema multiprocesador de que cualquier proceso (o hilo) se puede ejecutar en cualquier procesador; esto incluye procesos y código del núcleo. Una arquitectura SMP plantea nuevos elementos de diseño de los sistemas operativos y ofrece mayores rendimientos que un sistema monoprocesador en condiciones similares

En los últimos años, se ha prestado mucha atención al enfoque de los micronúcleos en el diseño de sistemas operativos. En su forma pura, un sistema operativo de micronúcleo está formado por un micronúcleo muy pequeño que ejecuta en modo de núcleo y que contiene sólo las funciones del sistema operativo más esenciales y críticas. El resto de las funciones del sistema operativo están implementadas para ejecutarse en modo de usuario y para usar el micronúcleo para los servicios críticos. El diseño con micronúcleos está dirigido a una implementación flexible y altamente modular. Sin embargo, aún existen dudas sobre el rendimiento de esta arquitectura.

Cuestiones de Repaso:

4.1. La Tabla 3.5 enumera los elementos más habituales de un bloque de control de proceso para un sistema operativo sin hilos. De ellos, ¿cuáles deberán pertenecer a un bloque control de hilo y cuáles a un bloque de control de proceso en un sistema multihilo?

En un entorno multihilo, un proceso se define como la unidad de protección y unidad de asignación de recursos. A los procesos se les asocian los siguientes elementos:

- Un espacio de direcciones virtuales, que contiene la imagen del proceso.
- Acceso protegido a los procesadores, otros procesos (para la comunicación entre procesos), archivos y recursos de E/S (dispositivos y canales).

En un proceso puede haber uno o más hilos, cada uno con lo siguiente:

- El estado de ejecución del hilo (Ejecución, Listo, etc.).
- El contexto del procesador, que se salva cuando no está ejecutando; una forma de ver el hilo es como un contador de programa independiente operando dentro de un proceso.
- Una pila de ejecución.
- Almacenamiento estático para las variables locales.
- Acceso a la memoria y a los recursos del proceso, compartidos con todos los otros hilos del mismo. (Página 151)

4.2. Indique razonadamente por qué un cambio de modo entre hilos puede ser menos costo que un cambio de modo entre procesos.

Porque no implica un costoso cambio de contexto.

4.3. ¿Cuáles son las dos características diferentes y potencialmente independientes que expresan el concepto de proceso?

- Unidad de propiedad de los recurso
- Unidad de Expedición (página 150)

4.4.Indique cuatro ejemplos generales del uso de hilos en un sistema monousuario multiprogramado.

- Trabajo interactivo y en segundo plano.
- Procesamiento asíncrono
- Aceleración de la ejecución
- Estructuración modular de los programas (página 153)

4.5.¿Qué recursos comparten, normalmente, los hilos de un proceso?

(...) todos los hilos de un proceso comparten el estado y los recursos del proceso. Residen en el mismo espacio de direcciones y tienen acceso a los mismos datos. (...) (página 152)

4.6.Enumere tres ventajas de los ULT frente a los KLT.
Son vanas las ventajas de usar ULT en lugar de KLT, entre otras están las siguientes:

1. El intercambio de hilos no necesita los privilegios del modo de núcleo, porque todas las estructuras de datos de gestión de hilos están en el espacio de direcciones de usuario de un mismo proceso. Por lo tanto, el proceso no debe cambiar a modo de núcleo para gestionar hilos. Con ello se evita la sobrecarga de dos cambios de modo (de usuario a núcleo y de núcleo a usuario).
2. Se puede realizar una planificación específica. Para una aplicación puede ser mejor la planificación mediante turno rotatorio mientras que para otra puede ser mejor la planificación por prioridades. Se puede realizar un algoritmo de planificación a medida de la aplicación sin afectar a la planificación subyacente del sistema operativo.
3. Los ULT pueden ejecutar en cualquier sistema operativo. Para dar soporte a ULT no es necesario realizar cambios en el núcleo subyacente. La biblioteca de hilos es un conjunto de utilidades de aplicación compartidas por todas las aplicaciones.
(página 159)

4.9. Defina brevemente las distintas arquitecturas nombradas en la Figura 4.8.

- **- Flujo de instrucción simple/dato simple (SISD):** un único procesador ejecuta un único flujo de instrucciones para operar sobre datos almacenados en una única memoria.
- **- Flujo de instrucción simple/datos múltiples (SIMD):** una única instrucción de máquina controla la ejecución simultánea de varios elementos del proceso según una secuencia de bloqueos. Cada elemento del proceso tiene una memoria de datos asociada, por lo que cada instrucción se ejecuta sobre un conjunto de datos diferente por medio de distintos procesadores. En esta categoría se encuentran los vectores y matrices de procesadores.
- **- Flujo de instrucción múltiple/dato simple (MISD):** se transmite una secuencia de datos a un conjunto de procesadores, cada uno de los cuales ejecuta una instrucción de la secuencia. Esta estructura no se ha implernentado nunca.
- **- Flujo de instrucción múltiple/datos múltiples (MIMD):** un conjunto de procesadores eje-cuta simultáneamente varias secuencias de instrucciones sobre distintos conjuntos de datos. (página 164)

4.10. Enumere los elementos clave de diseño para un sistema operativo SMP.
- Procesos e hilos concurrentes
- Planificación
- Sincronización
- Gestión de Memoria
- Fiabilidad y Tolerancia a fallos (página 166)

4.11.1. Indique ejemplos de funciones y servicios de un sistema operativo monolítico convencional que puedan ser subsistemas externos en un sistema operativo con micronúcleo.
Procesos Cliente, Gestores de dispositivos, Servidores de archivos, servidores de procesos, memoria virtual. (página 168)

4.12. Enumere y explique brevemente siete ventajas potenciales de un diseño con micronúcleo frente a un diseño monolítico.
- interfaz uniforme
- extensibilidad
- flexibilidad
- portabilidad
- fiabilidad
- soporte a sistemas distribuidos
- sistema operativo orientado a objetos (página 169)

4.13. Explique la desventaja potencial del rendimiento de un sistema operativo con micronúcleo.
(...) Una desventaja potencial, citada con frecuencia, de los micronúcleos es su rendimiento. Consume mas tiempo construir y enviar un mensaje, por aceptar y decodificar la respuesta, a través del micronúcleo que mediante una simple llamada al sistema. (...) (Página 170)

4.14. Enumere tres funciones que esperaría encontrar incluso en un sistema operativo con un micronúcleo mínimo.
(...) Un micronúcleo debe incluir aquellas funciones que dependen directamente del hardware, y cuya funcionalidad cuya funcionalidad es necesaria para dar soporte alas aplicaciones y servidores que ejecutan en modo núcleo. Estas funciones se engloban en las siguientes categorías generales: gestión de memoria de bajo nivel, comunicación entre procesos, gestión de interrupciones y E/S. (...) (página 171)

4.15. ¿Cuál es la forma básica de comunicación entre procesos o hilos en un sistema operativo con micronúcleo?
El pasaje de mensajes.

Capítulo 5 -	Concurrencia, exclusión mutua y sincronización

Resumen:

Los temas centrales de los sistemas operativos modernos son la multiprogramación, el multiproceso y el procesamiento distribuido. Un punto fundamental en estos temas y en las tecnologías de diseño de sistemas operativos es la concurrencia. Cuando se ejecutan varios procesos concurrente-mente, en el caso real de un sistema multiprocesador o en el caso virtual de un sistema monoprocesador multiprogramado, aparecen problemas de resolución de conflictos y de cooperación.

Los procesos concurrentes pueden interactuar de varias formas. Los procesos que no tienen Conocimiento unos de otros pueden competir por recursos tales como el tiempo del procesador o los dispositivos de E/S. Los procesos pueden tener conocimiento indirecto de los otros porque comparten el acceso a unos objetos comunes, como un bloque de memoria principal o un archivo. Finalmente, los procesos pueden tener un conocimiento directo de los otros y cooperar mediante intercambio de información. Los puntos clave que surgen en esta interacción son la exclusión mutua y el interbloqueo.

La exclusión mutua es una condición en la cual hay un conjunto de procesos concurrentes y sólo uno puede acceder a un recurso dado o realizar una función dada en cada instante de tiempo. Las técnicas de exclusión mutua pueden usarse para resolver conflictos tales como la competencia por los recursos y para sincronizar procesos de modo que puedan cooperar. Un ejemplo de esto último es el modelo del productor/consumidor, en el que un proceso pone datos en un buffer y uno o más procesos los extraen.

Se han desarrollado varios algoritmos en software para ofrecer exclusión mutua, de los cuales el más conocido es el algoritmo de Dekker. Las soluciones por software suelen tener un alto coste y el riesgo de errores lógicos en el programa es también alto. Un segundo conjunto de métodos para soportar la exclusión mutua suponen el uso de instrucciones especiales de la máquina. Estos métodos reducen la sobrecarga, pero son aún ineficientes porque emplean espera activa.

Otro método para dar soporte a la exclusión mutua consiste en incluir las características dentro del sistema operativo. Dos de las técnicas más comunes son los semáforos y el paso de mensajes. Los semáforos se usan para la señalización entre procesos y pueden emplearse fácilmente para hacer respetar una disciplina de exclusión mutua. Los mensajes son útiles para el cumplimiento de la exclusión mutua y ofrecen también un medio efectivo de comunicación entre procesos.

Cuestiones de Repaso:

5.1. Enumere cuatro elementos de diseño para los cuales es necesario el concepto de concurrencia.

- comunicación entre procesos
- compartición y competencia por los recursos
- sincronización en la ejecución de los procesos
- asignación de tiempo de procesador a los procesos (página 192)

5.2. ¿En qué tres contextos se presenta la concurrencia?

- Múltiples aplicaciones
- Aplicaciones estructuradas
- Estructura del sistema operativo (página 192)

5.3. ¿Cuáles son los requisitos básicos para la ejecución de procesos concurrentes?

(...) Se encontrará que la exigencia básica para soportar la concurrencia de procesos es la posibilidad de hacer cumplir la exclusión mutua, es decir, prohibir a los demás procesos realizar una acción cuando un proceso haya obtenido el permiso. (…) (Página 192)

Definición: Dos sentencias cualesquiera S1 y S2, pueden ejecutarse concurrentemente produciendo el mismo resultado si que si se ejecutasen secuencialmente si y solo si se cumplen las siguientes condiciones:

1) R(S1) intersección W(S2) = vacio
2) W(S2) intersección R(S1) = vacio
3) W(S1) intersección W (S2) = vacio

(W(Sx) = escritores de Sx; R(Sx) = lectores de Sx) (notas, tomo 1, página 192)

5.4. Enumere tres niveles de conocimiento entre procesos y defina brevemente cada uno de ellos.

- **Los procesos no tienen conocimiento de los demás**: estos son procesos independientes que no están pensados para operar juntos. El mejor ejemplo de esta situación es la multiprogramación de varios procesos independientes. Estos pueden ser tanto trabajos por lotes como sesiones interactivas o una combinación de ambos. Aunque los procesos no trabajen juntos, el sistema operativo tiene que encargarse de la competencia por los recursos. Por ejemplo, dos aplicaciones independientes pueden querer acceder al mismo disco, archivo o impresora. El sistema operativo debe regular estos accesos.

- **Los procesos tienen un conocimiento indirecto de los otros:** los procesos no conocen necesariamente a los otros por sus identificadores de proceso, pero comparten el acceso a algunos objetos, como un buffer de E/S. Estos procesos muestran cooperación para compartir el objeto común.

- **Los procesos tienen un conocimiento directo de los otros:** los procesos son capaces de comunicarse con los demás por el identificador de proceso y están diseñados para trabajar conjuntamente en alguna actividad. Estos procesos también muestran cooperación. (página 197)

5.5. ¿Cuál es la diferencia entre procesos en competencia y procesos en cooperación'?
Los procesos en competencia por los recursos no tienen conocimiento entre si, y los procesos en cooperación tienen conocimiento entre si y cooperan por compartimiento o por comunicación.

5.6. Enumere los tres problemas de control asociados a la competencia entre procesos y defina brevemente cada uno de ellos.

- Necesidad de exclusión mutua: que dos procesos no accedan simultáneamente a un recurso crítico.
- Interbloqueo: que los procesos tengan posesión de recursos que otros procesos necesitan.
- Inanición: que un proceso nunca llegue a obtener el control de los recursos que necesita.

5.7. Enumere los requisitos para la exclusión mutua.
Cualquier servicio o capacidad que dé soporte para la exclusión mutua debe cumplir los requisitos siguientes:

1. Debe cumplirse la excusión mutua: sólo un proceso de entre todos los que poseen secciones críticas por el mismo recurso u objeto compartido, debe tener permiso para entrar en ella en un instante dado.
2. Un proceso que se interrumpe en una sección no crítica debe hacerlo sin interferir con los otros procesos.
3. Un proceso no debe poder solicitar acceso a una sección crítica para después ser demorado indefinidamente; no puede permitirse el interbloqueo o la inanición.
4. Cuando ningún proceso está en su sección crítica, cualquier proceso que solicite entrar en la suya debe poder hacerlo sin dilación.
5. No se deben hacer suposiciones sobre la velocidad relativa de los procesos o el número de procesadores.
6. Un proceso permanece en su sección crítica sólo por un tiempo finito.

5.8. ¿Qué operaciones se pueden realizar sobre un semáforo'?
Para lograr el efecto deseado, se pueden contemplar los semáforos como variables que tienen un valor entero sobre el que se definen las tres operaciones siguientes:

1. Un semáforo puede iniciarse con un valor no negativo.
2. La operación w*ait* disminuye el valor del semáforo. Si el valor se hace negativo, el Proceso que ejecuta el w*ait* se bloquea. (Página 209)
3. La operación *signal* incrementa el valor del semáforo Si el valor no es positivo, se desbloquea un proceso bloqueado por una operación wait.

5.9. ¿Cuál es la diferencia entre los semáforos generales y los binarios?
Los semáforos binarios solo pueden tomar los valores 0 y 1.

5.10. ¿Cuál es la diferencia entre los semáforos débiles y los robustos?
Los semáforos robustos incluyen la política FIFO para la ejecución de los procesos, los débiles no.

5.11. ¿Qué es un monitor?
Los monitores son estructuras de un lenguaje de programación que ofrecen una funcionalidad equivalente a los semáforos y son más fáciles de controlar. (Página 225)

5.12. ¿Cuál es la diferencia entre *bloqueador y no bloqueador* con relación a los mensajes?
Un mensaje bloqueador es aquel que recibe un proceso que estaba bloqueado esperándolo.
Un mensaje no bloqueador es aquel que recibe un proceso que no estaba bloqueado esperándolo.

4.13. ¿Cuáles son las condiciones asociadas, en general, con el problema de los lectores/escritores'?

1) Cualquier número de lectores puede leer un archivo simultáneamente
2) Sólo puede escribir el archivo un escritor por vez
3) Cuando un escritor está accediendo al archivo ningún lector puede leerlo.

NOTA:
Otros temas importantes de éste capítulo:
Soluciones por software para la exclusión mutua: algoritmos de Dekker y Peterson.

Código de los semáforos.

```
struct semaforo
    mt contador;
    tipoCola cola;
void wait(semaforo s)
    s.contador--;
    ¡f (s.contador < 0)

poner este proceso en s.cola;
        bloquear este proceso;


void signal(semaforo s)

s.contador4 ~;
    ji (s.contador< =0)

      quitar un proceso P do s.cola;
      poner el proceso P en la cola de listos;
```

Figura 5.6. Una definición de las primitivas de los semáforos.

```
struct semaforo~binario
            enum (cero, uno) valor;
             tipoCola cola;
void waitB(semaforo~binario s)
    it (s.valor =       1)
      s.valor           0;
    else

        poner este proceso en s.cola;
        bloquear este proceso;


void signalB(semaforo s)

    it (s.cola.esvacia(>)
      s.valor   1;
    else

      quitar un proceso P de s.cola;
      poner el proceso P en la cola de listos;
```

Figura 5.7. Una definición de las primitivas de los semáforos binarios.

Capítulo 6 -	Concurrencia, interbloqueo e inanición

Resumen:

El interbloqueo es el bloqueo de un conjunto de procesos que compiten por los recursos del sistema o bien se comunican unos con otros. El bloqueo es permanente hasta que el sistema operativo realice alguna operación extraordinaria, como puede ser matar uno o más procesos u obligar a uno o más procesos a retroceder en la ejecución. El interbloqueo puede involucrar a recursos reutilizables o consumibles. Un recurso consumible es aquél que se destruye al ser adquirido por un proceso; como ejemplos se incluyen los mensajes y la información de los buffers de E/S. Un recurso reutilizable es aquél que no se agota o se destruye por el uso, como un canal de E/S o una zona de memoria.

Los métodos generales para hacer frente al interbloqueo son tres: prevención, detección y predicción. La prevención del interbloqueo garantiza que no se producirá el interbloqueo asegurando que no se cumpla ninguna de las condiciones necesarias para el interbloqueo. La detección del interbloqueo es necesaria si el sistema operativo está siempre dispuesto a conceder las peticiones de recursos; periódicamente, el sistema operativo debe comprobar la situación de interbloqueo y tomar medidas para deshacerlo. La predicción del interbloqueo supone el análisis de cada nueva petición de recursos para determinar si ésta puede conducir a un interbloqueo y concederlas sólo si no es posible llegar a un interbloqueo.

Cuestiones de Repaso:

6.1. Enumere ejemplos de recursos consumibles y reutilizables.

Reutilizables: procesadores, canales de E/S, memoria principal y secundaria, archivos, bases de datos, semáforos.
Consumibles: interrupciones, señales, mensajes, información en los buffers de E/S.

6.2. ¿Cuáles son las tres condiciones que deben darse para que sea posible el interbloqueo?

1 - Exclusión mutua
2 - Retención y espera
3 - No apropiación

6.3. ¿Cuáles son las cuatro condiciones que dan lugar al interbloqueo?

1 - Exclusión mutua
2 - Retención y espera
3 - No apropiación
4 - Circulo vicioso de espera

6.4. ¿Cómo se puede prevenir la condición de retener y esperar?

La condición de retención y espera puede prevenirse exigiendo que todos los procesos soliciten todos los recursos al mismo tiempo y bloqueando el proceso hasta que todos los recursos puedan concederse simultáneamente. (página 263)

6.5. Enumere dos formas en las que se puede prevenir la condición de no apropiación.

La condición de no apropiación pude prevenirse de varias formas. Primero, si a un proceso que retiene ciertos recursos se le deniega una nueva solicitud, dicho proceso deberá liberar todos sus recursos anteriores y solicitarlos de nuevo, cuando sea necesario, junto con el recurso adicional. Por otra parte, si un proceso si un proceso solicita un recurso retenido por otro proceso, el sistema operativo puede expulsar al segundo proceso y exigirle que libere sus recursos. Este último esquema evitará el interbloqueo sólo si no hay dos procesos que posean la misma prioridad. (Página 264)

6.6. ¿Cómo se puede prevenir la condición de círculo vicioso de espera'?

La condición del círculo vicioso de espera puede prevenirse definiendo una ordenación lineal de los tipos de recursos. Si a un proceso se le han asignado recursos de tipo R, entonces sólo podrá realizar peticiones posteriores sobre recursos de tipo siguiente de R en la ordenación. (Página 264)

6.7. ¿Cuál es la diferencia entre predicción, detección y prevención del interbloqueo?

En la prevención del interbloqueo, se obliga a las solicitudes de recursos a impedir que suceda, por lo menos, una de las cuatro condiciones. (...)

Con predicción del interbloqueo se pueden alcanzar las tres condiciones necesarias, pero se realizan las elecciones acertadas para no llegar al punto de interbloqueo (...) (página 264)

(...) con detección del interbloqueo se concederán los recursos a los procesos que lo necesiten siempre que sea posible. Periódicamente, el SO ejecutará un algoritmo que permite detectar la condición de circulo vicioso de espera (...) (página 270)

Nota:

También importante de éste capitulo: Algoritmo del banquero (predicción del interbloqueo), estado seguro e inseguro. (página 266)
- Algoritmo de detección de interbloqueo. (página 270)
- Cena de los filósofos (página 272)

Capítulo 7 -	Gestión de memoria

Resumen:

Una de las tareas más importantes y complejas de un sistema operativo es la gestión de memoria. La gestión de memoria implica tratar la memoria principal como un recurso que asignar y compartir entre varios procesos activos. Para un uso eficiente del procesador y de los servicios de E/S, resulta interesante mantener en la memoria principal tantos procesos como sea posible. Además, es deseable poder liberar a los programadores de las limitaciones de tamaño en el desarrollo de los programas.
Las herramientas básicas de la gestión de memoria son la paginación y la segmentación. En la paginación, cada proceso se divide en páginas de tamaño constante y relativamente pequeño. La segmentación permite el uso de partes de tamaño variable. También es posible combinar la segmentación y la paginación en un único esquema de gestión de memoria.

Cuestiones de Repaso:

7.1. ¿Cuáles son los requisitos que debe intentar satisfacer la gestión de memoria'?

- Reubicación
- Protección
- Compartición
- Organización física
- Organización lógica (página 292)

7.2. ¿Por qué es deseable la capacidad de reubicación?
Para poder cargar procesos en lugares diferentes de la memoria.

7.3. ¿Por qué no es posible implantar la protección de memoria en tiempo de compilación?
Porque no se conoce de antemano la dirección física que el proceso ocupará en tiempo de ejecución.

7.4. ¿Cuáles son algunas de las razones para permitir a dos o más procesos tener acceso a una región de memoria en panicular?
Para poder compartir estructuras de datos entre diferentes procesos.

7.5. En un esquema de partición estática, ¿cuáles son las ventajas de usar particiones de distinto tamaño?
Reducir la fragmentación interna. Proporcionar cierta flexibilidad mayor que la obtenida en el esquema de partición estática con tamaños iguales.

7.6. ¿Cuál es la diferencia entre la fragmentación interna y la externa?
Fragmentación interna: cuando un proceso ocupa menos espacio que el asignado a él por el sistema operativo.
Fragmentación externa: Cuando en la memoria quedan huecos no utilizados entre procesos que la ocupan.

7.7. ¿Cuáles son las diferencias entre direcciones lógicas, relativas y físicas?
Una **dirección lógica** es una referencia a una posición de memoria independiente de la asignación actual de datos a la memoria; se debe hacer una traducción a una dirección física antes de poder realizar un acceso a la memoria. Una **dirección relativa** es un caso particular de dirección lógica, en el cual la dirección se expresa como una posición relativa a algún punto conocido, normalmente el principio del programa. Una **dirección física o** dirección absoluta, es una posición real en la memoria principal.
(Página 305)

7.8. ¿Cuál es la diferencia entre una página y un marco de página?
(...) los trozos del proceso, llamados **páginas,** pueden asignarse a los trozos libres de memoria, llamados marcos o marcos de página. El término marco o encuadre *(frame)* se utiliza porque un marco puede mantener o encuadrar una página de datos.(...)
(página 306)

7.9. ¿Cuál es la diferencia entre página y segmento?
Los segmentos pueden ser de tamaño variable y las páginas son de tamaño fijo.

Nota:

También importante BUDDY SYSTEM

Capítulo 8 -	Memoria Virtual

Resumen:

Para un aprovechamiento eficiente del procesador y de los servicios de E/S es conveniente mantener en la memoria principal tantos procesos como sea posible. Además, conviene liberar a los programadores de las limitaciones de tamaño en el desarrollo de programas.

La forma de abordar ambos problemas es por medio de la memoria virtual. Con memoria virtual, todas las referencias a direcciones son referencias lógicas que se traducen a direcciones reales durante la ejecución. Esto permite a los procesos situarse en cualquier posición de la memoria principal y cambiar de ubicación a lo largo del tiempo. La memoria virtual permite también dividir un proceso en fragmentos. Estos fragmentos no tienen por qué estar situados de forma contigua en la memoria principal durante la ejecución y no es ni siquiera necesario que todos los fragmentos del proceso estén en la memoria durante la ejecución.

Los dos enfoques básicos de memoria virtual son la paginación y la segmentación. Con la paginación, cada proceso se divide en páginas de tamaño fijo y relativamente pequeño. La segmentación permite el uso de fragmentos de tamaño variable. También es posible combinar segmentación y paginación en un único esquema de gestión de memoria.

Un esquema de gestión de memoria virtual exige un soporte tanto de hardware como de software. El soporte de hardware lo proporciona el procesador. Este soporte incluye la traducción dinámica de direcciones virtuales a direcciones físicas y la generación de interrupciones cuando una página o segmento referenciado no están en la memoria principal. Estas interrupciones activan el software de gestión de memoria del sistema operativo.

Una serie de cuestiones de diseño relativas a los sistemas operativos dan soporte a la gestión de memoria virtual:

*** Políticas de lectura:** las páginas de los procesos pueden cargarse por demanda o se puede usar una política de paginación previa; esta última agrupa las actividades de entrada cargan-do varias páginas a la vez.

*** Políticas de ubicación:** en un sistema de segmentación pura, un segmento entrante debe encajar en un espacio de memoria disponible.

*** Políticas de reemplazo:** cuando la memoria está llena, debe tomarse la decisión de qué página o páginas serán reemplazadas.

*** Gestión del conjunto residente:** el sistema operativo debe decidir cuánta memoria principal ha de asignar a un proceso en particular cuando se carga. Puede hacerse una asignación estática en el momento de la creación del proceso o bien puede cambiar dinámicamente.

*** Políticas de vaciado:** las páginas de un proceso modificadas pueden expulsarse al disco en el momento del reemplazo o bien puede aplicarse una política de vaciado previo; esta última agrupa la actividad de salida expulsando varias páginas de una vez.

*** Control de carga:** el control de carga determina el número de procesos residentes que habrá en la memoria principal en un momento dado.

Cuestiones de Repaso:

8.1. ¿Cuál es la diferencia entre paginación simple y paginación con memoria virtual?

En el estudio de la paginación simple se indicó que cada proceso tiene su propia tabla de páginas y que. cuando carga todas sus páginas en la memoria principal, se crea y carga en la memoria principal tina tabla de páginas. Cada entrada de la tabla de páginas contiene el número de marco de la página correspondiente en la memoria principal. Cuando se considera un esquema de memoria virtual basado en la paginación se necesita la misma estructura, una tabla de páginas. Nuevamente, es normal asociar una única tabla de páginas con cada proceso. En este caso, sin embargo, las entradas de la tabla de páginas pasan a ser más complejas. Puesto que sólo algunas de las páginas de un proceso pueden estar en la memoria principal, se necesita un bit en cada entrada de la tabla para indicar si la página correspondiente está presente (P) en la memoria principal o no lo está. Si el bit indica que la página está en la memoria, la entrada incluye también el número de marco para esa página.

Otro bit de control necesario en la entrada de la tabla de páginas es el bit de modificación (M), para indicar si el contenido de la página correspondiente se ha alterado desde que la página se cargó en la memoria principal. Si no ha habido cambios, no es necesario escribir la página cuando sea sustituida en el marco que ocupa actualmente. Puede haber también otros bits de control. Por ejemplo, si la protección o la compartición se gestiona en la página. se necesitarán más bits con tal propósito.

(Páginas 328 y 329)

8.2. Explique la hiperpaginación.

(...) Demasiados intercambios de fragmentos conducen a lo que se conoce como hiperpaginación (thrashing): el procesador consume mas tiempo intercambiando fragmentos que ejecutando instrucciones de usuario. (...) (página 325)

8.3. ¿Por qué es el principio de cercanía crucial para el uso de la memoria virtual'?

Los argumentos anteriores se basan en el **principio de cercanía,** que se introdujo en el Capítulo 1 (véase especialmente el Apéndice lA). Resumiendo, el principio de cercanía afirma que las referencias a los datos y al programa dentro de un proceso tienden a agruparse. Por lo tanto, es válida la suposición de que, durante cortos periodos de tiempo, se necesitarán sólo unos pocos fragmentos de un proceso. Además, sería posible hacer predicciones inteligentes sobre qué fragmentos de un proceso se necesitarán en un futuro cercano y así evitar la hiperpaginación. (Página 326)

8.4. ¿Qué elementos se encuentran, normalmente, en una entrada de tabla de páginas?

Defina brevemente cada uno de ellos.

Así pues, el mecanismo básico de lectura de una palabra de la memoria supone la traducción por medio de la tabla de páginas de una dirección virtual o lógica, formada por un número de página y un desplazamiento, a una dirección física que está formada por un número de marco y un desplazamiento. Puesto que la tabla de páginas es de longitud variable, en función del tamaño del *proceso*, no es posible suponer que quepa en los registros. En su lugar, debe estar en la memoria principal para ser accesible. La Figura 8.3 sugiere una implementación en hardware de este esquema. Cuando se está ejecutando un proceso en particular, la dirección de comienzo de la tabla de páginas para este proceso se mantiene en un registro. El número de página de la dirección virtual se emplea como índice en esta tabla para buscar el número de marco correspondiente. Este se combina con la parte de desplazamiento de la dirección virtual para generar la dirección real deseada. (página 329)

8.5. ¿Cuál es el propósito del buffer de traducción adelantada?
En principio, cada referencia a la memoria virtual puede generar dos accesos a la memoria: uno para obtener la entrada de la tabla de páginas correspondiente y otro para obtener el dato deseado. Así pues, un esquema sencillo de memoria virtual podría tener el efecto de doblar el tiempo de acceso a la memoria. Para solucionar este problema, la mayoría de los esquemas dc memoria virtual hacen Liso de una cache especial para las entradas de la tabla de páginas, llamada generalmente **buffer de traducción adelantada(...) (página 332)**

8.6. Defina brevemente las alternativas en políticas de lectura de páginas.
Con la **paginación por demanda, se** trae una página a la memoria principal sólo cuando se hace referencia a una posición en dicha página. Si los otros elementos de la política de gestión de memoria funcionan adecuadamente, debe ocurrir lo siguiente: cuando un proceso se ejecute por **primera** vez, se producirá un aluvión de fallos de página. A medida que se traigan a la memoria más páginas, el principio de cercanía hará que la mayoría de las futuras referencias estén en páginas que se han cargado hace poco. Así pues, después de un tiempo, la situación se estabilizará y el número de fallos de página disminuirá hasta un nivel muy bajo.
Con la paginación **previa,** se cargan otras páginas distintas a las demandadas debido a un fallo de página. El principal atractivo de esta estrategia está en las características de la mayoría de los dispositivos de memoria secundaria, como los discos, que tienen un tiempo de búsqueda y una latencia de giro. Si las páginas de un proceso se cargan secuencialmente en la memoria secundaria, es mas eficiente traer a la memoria un número de páginas contiguas de una vez que ir trayéndolas (le una en una durante un periodo largo de tiempo. Por supuesto, esta política no es efectiva si la mayoría (le las páginas extra que se traen no se referencian. (página 343)

8.7. ¿Cuál es la diferencia entre gestión del conjunto residente y política de reemplazo de páginas?
La política de reemplazo de páginas se encarga de seleccionar la página a reemplazar entre las que están actualmente en la memoria.
La gestión del conjunto residente es lo contrario, decide cuales páginas se van a cargar en la memoria principal.

8.8. ¿Cuál es la relación entre los algoritmos de reemplazo de páginas FIFO y del reloj'?
Los algoritmos son similares, pero el algoritmo del reloj incluye un bit se señalización para eliminar a las páginas más antiguas de la memoria.

8.9. ¿Cuál es la ventaja del almacenamiento intermedio de páginas?
(...)Lo importante de estas operaciones es que la página a reemplazar permanece en la memoria. Así pues, si el proceso hace referencia a dicha página, se devuelve al conjunto residente del proceso con un coste pequeño. En realidad, las listas de páginas libres y modificadas actúan como una cache dc páginas. La lista de páginas modificadas tiene otra función provechosa: las páginas modificadas son reescritas por bloques, en vez de una a una. Esto reduce significativamente el número de operaciones de E S y por lo tanto, la cantidad de tiempo de acceso al disco. (...) (Página 350)

8.10. ¿Por qué no es posible combinar una política de reemplazo global y una política de asignación fija?
Porque la política de asignación fija implica que los procesos se carguen en un número fijo de páginas en las que ejecutar y la política de reemplazo global considera a todas las páginas de la memoria como candidatas a reemplazar, independientemente del proceso al que pertenezcan.

8.11. ¿Cuál es la diferencia entre un conjunto residente y un conjunto de trabajo?
Un conjunto de trabajo es un espacio virtual al que el proceso ha hecho referencia en un tiempo determinado y un conjunto residente es el conjunto total de páginas en memoria del proceso.

8.12. ¿Cuál es la diferencia entre vaciado por demanda y vaciado previo?
Vaciado por demanda: la página se descarga a memoria secundaria sólo cuando haya sido elegida para reemplazarse.
Vaciado previo: la página que ha sido se escribe en memoria secundaria antes que se necesite su marco.

Capítulo 9 -	Planificación de monoprocesadores

Resumen:

El sistema operativo puede tomar tres tipos de decisiones de planificación que afectan a la ejecución de los procesos. La planificación a largo plazo determina cuándo se admiten nuevos procesos en el sistema. La planificación a medio plazo forma parte de la función de intercambio y determina cuándo se lleva parcial o totalmente un proceso a la memoria principal para que pueda ser ejecutado. La planificación a corto plazo determina cuál de los procesos listos será ejecutado a continuación por el procesador. Este capítulo se centra en los asuntos relativos a la planificación a corto plazo.

En el diseño de un planificador a corto plazo se emplean gran variedad de criterios. Algunos de estos criterios hacen referencia al comportamiento del sistema tal y como lo percibe el usuario(orientados a usuario), mientras que otros consideran la efectividad total del sistema para satisfacer las necesidades de todos los usuarios (orientados al sistema). Algunos de los criterios se refieren concretamente a medidas cuantitativas del rendimiento, mientras que otros son de tipo cualitativo. Desde el punto de vista del usuario, la característica más importante de un sistema es, en general, el tiempo de respuesta, mientras que desde el punto de vista del sistema es más importante la productividad o la utilización del procesador.
Se ha desarrollado una gran variedad de algoritmos para tomar las decisiones de planificación a corto plazo entre los procesos listos. Entre estos se incluyen:

- **Primero en llegar/primero en servirse:** selecciona el proceso que lleva más tiempo esperando servicio.
- **Turno rotatorio:** emplea un fraccionamiento del tiempo para hacer que los procesos se limiten a ejecutar en ráfagas cortas de tiempo, rotando entre los procesos listos.
- **Primero el proceso más corto:** selecciona el proceso con menor tiempo esperado de ejecución, sin apropiarse de la CPU.
- **Menor tiempo restante:** selecciona el proceso al que le queda menos tiempo esperado de ejecución en el procesador. Un proceso puede ser expulsado cuando otro proceso está listo.
- **Primero la mayor tasa de respuesta:** la decisión de planificación se basa en una estimación del tiempo de retorno normalizado.
- **Realimentación:** establece un conjunto de colas de planificación y sitúa los procesos en las colas, teniendo en cuenta, entre otros criterios, el historial de ejecución.

La elección de un algoritmo de planificación dependerá del rendimiento esperado y de la complejidad de la implementación.

Cuestiones de Repaso:

9.1. Describa brevemente los tres tipos de planificación de procesador.
Largo Plazo: Decisión de añadir procesos al conjunto de procesos a ejecutar
Mediano Plazo: Decisión de añadir procesos al conjunto de procesos que se encuentran parcial o completamente en memoria
Corto Plazo: Decisión sobre que proceso será ejecutado por el procesador (página 384)

9.2. ¿Qué es habitualmente un factor de rendimiento crítico en un sistema operativo interactivo?
El tiempo de respuesta.

9.3. ¿Cuál es la diferencia entre Tiempo de Retorno y Tiempo de Respuesta?
Uno está orientado al usuario (respuesta) y otro a la terminación de un proceso (retorno).

9.4. En la planificación de procesos, ¿un valor de prioridad bajo representa una baja o alta prioridad?
Baja prioridad.

9.5. ¿Cuál es la diferencia entre planificación preferente y no preferente?
Preferente: (preemptive, con reemplazo en el uso de la CPU, o prendario) el proceso que se está ejecutando puede ser interrumpido y pasado al estado listo por el sistema operativo.
Round Robin, SRT (shortest remaining time), **HRRN** (high response ratio first, o primero el de mayor tasa de respuesta), Feedback (retroalimentación),
No preferente: (non preemptive, sin reemplazo, apropiativos o no prendario) una vez que el proceso ha llegado al estado de ejecución continúa ejecutándose hasta que termina o se bloquea en espera de E/S o una petición al sistema.
FIFO (o FCFS), **SPF** (o SPN: shortest procces next), por **PRIORIDAD**

9.6. Defina brevemente la planificación FCFS.
Selecciona el proceso que lleva más tiempo esperando servicio.

9.7. Defina brevemente la planificación por Turno Rotatorio (RR).
Emplea un fraccionamiento del tiempo para hacer que los procesos se limiten a ejecutar en ráfagas cortas de tiempo, rotando entre los procesos listos.

9.8. Defina brevemente la planificación por Primero el Proceso Más Corto (SPN).
Selecciona el proceso con menor tiempo esperado de ejecución, sin apropiarse de la CPU.

9.9. Defina brevemente la planificación por Menor Tiempo Restante (SRT).
Selecciona el proceso al que le queda menos tiempo esperado de ejecución en el procesador. Un proceso puede ser expulsado cuando otro proceso está listo.

9.10. Defina brevemente la planificación por Primero la Mayor Tasa de Respuesta (HRRN).
La decisión de planificación se basa en una estimación del tiempo de retorno normalizado.

9.11. Defina brevemente la planificación por Realimentación (FB).
Establece un conjunto de colas de planificación y sitúa los procesos en las colas, teniendo en cuenta, entre otros criterios, el historial de ejecución.

Capítulo 10 -	Planificación de multiprocesadores y en tiempo real

Resumen:

En un multiprocesador fuertemente acoplado, varios procesadores tienen acceso a la misma memoria principal. Con esta configuración, la estructura de planificación es algo más compleja. Por ejemplo, se puede asignar un determinado proceso al mismo procesador durante toda su vida o se puede expedir hacia un procesador distinto cada vez que alcance el estado Ejecutando. Algunos estudios de rendimiento proponen que las diferencias entre los diversos algoritmos de planificación son menos significativas en un sistema multiprocesador.

Un proceso o tarea de tiempo real es aquél que se ejecuta en conexión con algún proceso, función o conjunto de sucesos externos al sistema informático y que debe cumplir uno o más plazos para interactuar de forma correcta y eficiente con el entorno exterior. Un sistema operativo en tiempo real es aquél que gestiona procesos de tiempo real. En este contexto, no son aplicables los criterios tradicionales de selección de algoritmos de planificación. En su lugar, el factor clave está en cumplir los plazos. Son apropiados en este contexto los algoritmos que dependen mucho de la apropiación y de la reacción a los plazos relativos.

Cuestiones de Repaso:

10.1. Enumere y defina brevemente las cinco categorías de granularidad de sincronización.

Tabla 10.1. Procesos y granularidad de la sincronización.

Tamaño de grano	Descripción	Intervalo de sincronización (instrucciones)
Fino	Paralelismo inherente en un único flujo de instrucciones	< 20
Medio	Procesamiento paralelo o multitarea dentro de una aplicación individual	20-200
Grueso	Multiprocesamiento de procesos concurrentes en un entorno multiprogramado	200-2000
Muy grueso	Proceso distribuido por los nodos de una red para formar un solo entorno de computación	2000-1M
Independiente	Varios procesos no relacionados	(N/A)

(página 427)

10.2. Enumere y defina brevemente las cuatro técnicas de planificación de hilos.

Entre las diversas propuestas de planificación de hilos para multiprocesadores y de asignación de procesadores destacan los siguientes cuatro métodos:

- Reparto de carga: los procesos no se asignan a un procesador en particular Se mantiene una cola global de hilos listos y cada procesador, cuando está ocioso, selecciona un hilo de la cola. El termino *reparto de carga* se emplea para distinguir esta estrategia del esquema de balance de carga, en el que el trabajo se asigna de forma más permanente
- Planificación por grupos: se planifica un conjunto de hilos afines para su ejecución en un conjunto de procesadores al mismo tiempo, según una relación de uno a uno.
- Asignación dedicada de procesadores: es el enfoque opuesto al reparto de carga y ofrece una planificación implícita definida por la asignación de hilos a los procesadores. Mientras un programa se ejecuta, se le asigna un número de procesadores igual al número de hilos que posea. Cuando el programa finaliza, los procesadores retornan a la reserva general para posibles asignaciones a otros programas.
- Planificación dinámica: el número de hilos en un programa se puede cambiar en el curso de la ejecución.

(página 432)

10.3. Enumere y defina brevemente las tres versiones de reparto de carga.

- Primero en llegar/primero en servirse (FCFS): cuando llega un trabajo, cada uno de sus hilos se sitúa consecutivamente al final de la cola compartida. Cuando un procesador pasa a estar ocioso, toma el siguiente hilo listo y lo ejecuta hasta que finalice o se bloquee.
- Primero el de menor número de hilos: la cola de listos compartida se organiza como una cola de prioridades, en la que la prioridad más alta se asigna a los hilos de los trabajos con el menor número de hilos sin planificar. Los trabajos de la misma prioridad se ordenan según el orden de llegada. Como con FCFS, un hilo planificado se ejecuta hasta que finaliza o se bloquea.
- Primero el de menor número de hilos (preferente): la mayor prioridad se da a los trabajos con el menor número de hilos sin terminar. La llegada de un trabajo con un número de hilos menor que un trabajo en ejecución expulsará los hilos del trabajo planificado. (Página 433)

10.4. ¿Cuál es la diferencia entre tareas de tiempo real rígidas y flexibles?

Una **tarea rígida de tiempo real** debe cumplir el plazo; en otro caso producirá daños no deseados o un error fatal en el sistema. Una **tarea flexible de tiempo real** tiene un plazo asociado, que es conveniente, pero no obligatorio; aunque haya vencido el plazo, aún tiene sentido planificar y completar la tarea. (página 438)

10.5. ¿Cuál es la diferencia entre tareas de tiempo real periódicas y aperiódicas?

Una **tarea aperiódica** debe comenzar o terminar en un plazo o bien puede tener una restricción tanto para el comienzo como para la finalización. En el caso de una **tarea periódica,** el requisito se puede enunciar como «una vez por cada periodo *T*» o «exactamente cada *T* unidades». (página 438)

10.6. Enumere y defina brevemente las cinco áreas generales de requisitos para sistemas operativos en tiempo real.

- Determinismo: realizar operaciones en instantes fijos
- Sensibilidad: tiempo dl sistema para dar servicio a una interrupción
- Control del usuario: Permitir al usuario un control especifico sobre la prioridad de las tareas.
- Fiabilidad: medida de la calidad del sistema
- Tolerancia a fallos: característica que hace referencia a conservar la capacidad de respuesta en caso de fallos.

10.7. Enumere y defina brevemente las cuatro clases de algoritmos de planificación en tiempo real.

- **Métodos con tablas estáticas:** realizan un análisis estático de las planificaciones de expedición posibles. El resultado del análisis es una planificación que determina, en tiempo de ejecución, cuándo debe comenzar la ejecución de una tarea.
- **Métodos preferentes con prioridades estáticas:** también se realiza un análisis estático, pero no se realiza ninguna planificación. En cambio, se usa dicho análisis para asignar prioridades a tareas, de forma que se puede emplear un planificador convencional preferente con prioridades.
- Métodos **de planificación dinámica:** se determina la viabilidad en tiempo de ejecución (dinámicamente) en vez de antes de empezar la ejecución (estáticamente). Se acepta una nueva tarea para ejecutar sólo si es factible cumplir sus restricciones de tiempo. Uno de los resultados del análisis de viabilidad es un plan o proyecto empleado para decidir cuándo se expide cada tarea.
- **Métodos dinámicos del mejor resultado:** no se realiza ningún análisis de viabilidad. El sistema intenta cumplir todos los plazos y abandona cualquier proceso ya iniciado y cuyo plazo no se haya cumplido.

10.8. ¿Qué elementos de información de las tareas pueden ser útiles en la planificación en tiempo real?

- **Instante en que está lista:** el instante en que la tarea pasa a estar lista para ejecución. En el caso de una tarea repetitiva **o** periódica, es en realidad una secuencia de instantes conocidos con anterioridad. En el caso de una tarea aperiódica, este tiempo puede ser conocido con anterioridad o bien el sistema operativo puede tener conocimiento de él solamente cuando la tarea ya se encuentre lista.
- **Plazo de comienzo:** instante en el que la tarea debe comenzar.
- **Plazo de finalización:** instante en el que la tarea debe terminar. Las aplicaciones típicas de tiempo real tienen normalmente un plazo de comienzo o un plazo de finalización pero no ambos.
- **Tiempo de proceso:** el tiempo necesitado para ejecutar una tarea hasta su finalización. En algunos casos, este tiempo es facilitado, pero, en otros, el sistema operativo calcula una media exponencial. En otros sistemas de planificación, no se usa esta información.
- **Exigencias de recursos:** el conjunto de recursos (además del procesador) que necesita una tarea durante su ejecución.
- **Prioridad:** mide la importancia relativa de la tarea. Las tareas rígidas de tiempo real pueden tener una prioridad «absoluta», produciéndose un fallo del sistema si un plazo se pierde. Si el sistema continúa ejecutándose pase lo que pase, tanto las tareas rígidas de tiempo real como las flexibles recibirán una prioridad relativa como guía para el planificador.
- **Estructura de subtareas:** una tarea puede descomponerse en una subtarea obligatoria y otra subtarea opcional. Sólo la subtarea obligatoria tiene un plazo rígido.

Capítulo 11 -	Gestión de E/S y planificación de discos

Resumen:

La interfaz de un sistema informático con el mundo exterior es la arquitectura de E/S. Esta arquitectura está diseñada para ofrecer un medio sistemático de controlar la interacción con el mundo exterior y proporcionar al sistema operativo la información que necesita para administrar la actividad de E/S de una manera eficaz.

Las funciones de E/S se dividen generalmente en un conjunto de niveles, donde los más bajos se encargan de los detalles cercanos a las funciones físicas a realizar y los superiores tratan con la E/S desde un punto de vista lógico y general. El resultado es que los cambios en los parámetros del hardware no afectan necesariamente a la mayor parte del software de E/S.

Un aspecto clave de la E/S es el empleo de buffers controlados por utilidades de E/S más que por los procesos de aplicación. El almacenamiento intermedio sirve para igualar las diferencias de velocidades internas del sistema informático y las velocidades de los dispositivos de E/S. El uso de buffers también permite desacoplar las transferencias reales de E/S del espacio de direcciones del proceso de aplicación, lo que permite al sistema operativo una mayor flexibilidad en la realización de las funciones de gestión de memoria.

El aspecto de la E/S que tiene un mayor impacto en el rendimiento global del sistema es la E/S a disco. Por consiguiente, se han realizado más investigaciones e invertido más esfuerzos de diseño en este punto que en cualquier otro de la E/S. Dos de los métodos usados más frecuentemente para mejorar el rendimiento de la E/S a disco son la planificación y la cache de disco.

En un instante dado, puede haber una cola de solicitudes de E/S al mismo disco. Es una labor de la planificación del disco el satisfacer estas peticiones de forma que se minimice el tiempo de búsqueda mecánica del disco y, por tanto, se mejore el rendimiento. Aquí entran en juego la disposición física de las solicitudes pendientes, así como consideraciones sobre la cercanía de las mismas.

Una cache de disco es un buffer, normalmente en la memoria principal, que funciona como una cache de bloques de disco entre la memoria del disco y el resto de la memoria principal. Por el principio de cercanía, el empleo de una cache de disco debe reducir sustancialmente el número de transferencias de E/S de bloques entre la memoria principal y el disco.

Cuestiones de Repaso:

11.1. Enumere y defina brevemente las tres técnicas de realización de E/S.

- E/S programada: el procesador emite una orden de E/S de parte de nu proceso a un módulo de E/S; el proceso espera entonces que termine la operación antes de seguir.
- E/S dirigida por interrupciones: el procesador emite una orden de E/S de parte de un proceso, continúa la ejecución de las instrucciones siguientes y el módulo de E/S lo interrumpe cuando completa su trabajo. (...)
- DMA: un módulo de DMA controla en intercambio de datos entre la memoria principal y un módulo de E/S. (página 464)

11.2. ¿Cuál es la diferencia entre E/S lógica y E/S a dispositivo?
Uno se ocupa de las funciones generales de E/S solicitadas por los procesos de usuario (E/S Lógica) y el otro emitir las instrucciones adecuadas de E/S al dispositivo (E/S a dispositivo)

11.3. ¿Cuál es la diferencia entre un dispositivo orientado a bloque y un dispositivo orientado a flujo? Dé un ejemplo de cada uno de ellos.

Para el estudio de los distintos métodos de almacenamiento intermedio, a veces es importante hacer una distinción entre dos tipos de dispositivos: dispositivos orientados a bloque y dispositivos orientados a flujo. Los **dispositivos orientados a bloque** almacenan la información de bloques, normalmente de tamaño fijo, haciendo las transferencias de un bloque cada vez. Generalmente, es posible referirse a los (latos por su número de bloque. Los discos y las cintas son ejemplos de dispositivos orientados a bloques. Los dispositivos orientados a **flujo** transfieren los datos como una serie de bytes; no poseen estructura de bloques. Terminales, impresoras, puertos de comunicación, ratones y otros dispositivos apuntadores y la mayoría de los dispositivos restantes que no son de almacenamiento secundario son dispositivos orientados a flujos.
(Página 471)

11.4. ¿Por qué podría esperar una mejora del rendimiento utilizando para la E/S una memoria intermedia doble en vez de una sencilla?

Se puede realizar una mejora sobre la memoria intermedia sencilla asignando a la operación dos almacenes intermedios del sistema (Figura 1 1.6c). De esta forma, un proceso puede transferir datos hacia (o desde) una memoria intermedia mientras que el sistema operativo vacía (o rellena) (página 473)

11.5. ¿Cuáles son los retardos que intervienen en una lectura o escritura de disco'?

- Tiempo de búsqueda (seek time)
- Retardo de giro (demora de rotación o Rotacional Delay)
- Tiempo de transferencia (Transfer Time)

11.6. Defina brevemente las políticas de planificación de disco que ilustra la Figura 11.8

- FIFO: los pedidos de acceso se procesan en orden secuencial
- SSTF: (shortest scan time first) primero el de menor tiempo de servicio
- SCAN (look): el brazo atiende las solicitudes en un solo sentido hasta llegar al final, cuando cambia de dirección.
- C-SCAN: todas las solicitudes se satisfacen en una sola dirección (ascendente o descendente)
- (también importantes son: N-SCAN, F-SCAN, C-LOOK UP)

11.7. Defina brevemente los siete niveles RAID.

- Nivel 0: agrupa dos o mas discos físicos para formar un disco lógico, no podee redundancia de datos
- Nivel 1: espejo, copia toda la información del primer conjunto de discos en el segundo
- Nivel 2: redundancia por código de hamming, utiliza polinomios de hamming en discos adicionales para dar fiabilidad a los datos, requiere n/2 –1 discos redundantes, ésta técnica fue superada.
- Nivel 3: paridad por intercalación de bits, requiere un solo disco redundante.
- Nivel 4: paridad por intercalación de bloques, requiere un disco redundante, almacena la paridad pero calculada por bloque.
- Nivel 5: igual al nivel 4 salvo que distribuye la paridad entre todos los discos intercaladamente.
- Nivel 6: = al nivel 5 salvo que agrega otra forma de controlar la paridad independiente y requiere 2 discos de paridad.

11.8. ¿Cuál es el tamaño normal de un sector de disco?
512 bytes

Nota: Importante la forma de calcular el tiempo de acceso

Capítulo 12 -	Gestión de Archivos

Resumen:

Un sistema de gestión de archivos es el software del sistema que proporciona servicios a usuarios y aplicaciones para el uso de archivos, incluyendo el acceso a archivos, e conservación de directorios y control de acceso. Normalmente, el sistema de gestión de archivos se contempla como un servido del sistema (1UC SC sirve a su vez del sistema operativo, más que como una parte del propio sistema operativo. Sin embargo, en cualquier sistema, al menos una parte de las funciones de gestión de archivos las realiza el sistema operativo.

Un archivo es un conjunto de registros. La forma en que se accede a estos registros determina su organización lógica y, hasta cierto punto, su organización física en el disco. Si un archivo va a ser básicamente procesado en su totalidad, la organización secuencial es la más simple y adecuada. Si el acceso secuencial es necesario pero también se desea el acceso aleatorio al archivo, un archivo secuencial indexado puede dar el mejor rendimiento. Si el acceso al archivo es principalmente aleatorio, un archivo indexado o un archivo de dispersión puede ser el más apropiado.

Sea cual sea la estructura de archivo elegida, se necesita también un servicio de directorios. Este permite a los archivos organizarse de una forma jerárquica. Esta organización es útil para que el usuario siga la pista de los archivos y para que el sistema de gestión de archivos proporcione a los usuarios un control de acceso junto a otros servicios.

Los registros de archivos, incluso los de tamaño fijo, no se ajustan generalmente al tamaño del bloque físico del disco. De esta forma, se necesita algún tipo de estrategia de agrupación. La estrategia de agrupación que se use quedará determinada por un equilibrio entre la complejidad, el rendimiento y el aprovechamiento del espacio.

Una función clave de cualquier esquema de gestión de archivos es la gestión del espacio en el disco. Una parte de esta función es la estrategia de asignación de bloques de disco a los archivos. Se han empleado una amplia variedad de métodos y de estructuras de datos para guardar constancia de la ubicación de cada archivo. Además, también debe gestionarse el espacio en el disco sin asignar. Esta última función consiste principalmente en mantener una tabla de asignación de disco que indique los bloques que están libres.

Cuestiones de Repaso:

12.1. ¿Cuál es la diferencia entre un campo y un registro?
Un campo es un elemento básico de datos y un registro es una agrupación de campos.

12.2. ¿Cuál es la diferencia entre un archivo y una base de datos?
Un archivo es una agrupación de registros similares y una base de datos es un conjunto de datos relacionados, normalmente una base de datos está diseñada para ser usada por varias aplicaciones diferentes y un archivo no.

12.3. ¿Qué es un sistema de gestión de archivos?
Un sistema de gestión de archivos es un conjunto de software del sistema que ofrece a los usuarios y a las aplicaciones servicios relativos al empleo de archivos. (Página 515)

12.4. ¿Qué criterios son importantes en la elección de una organización de archivos?

- acceso rápido
- facilidad de actualización
- economía de almacenamiento
- mantenimiento sencillo
- fiabilidad (página 519)

12.5. Enumere y defina brevemente cinco organizaciones de archivos.

- pilas
- archivos secuenciales
- archivos secuenciales indexados
- archivos indexados
- archivos directos o de dispersión (hash) página 519

12.6. ¿Por qué es el tiempo medio de búsqueda de un registro menor en un archivo secuencial indexado que en un archivo secuencial?
Porque se accede directamente a través del índice.

12.7. ¿Cuáles son las operaciones típicas que se pueden realizar sobre un directorio'?

- buscar
- crear directorio
- borrar archivo
- Enumerar directorio
- Actualizar directorio (página 525)

12.8. ¿Cuál es la relación entre un nombre de ruta y un directorio de trabajo?
Un nombre de ruta termina con el nombre de un archivo y el directorio de trabajo es el que está asociado a ese archivo.

12.9. ¿Cuáles son los derechos de acceso típicos que se pueden conceder o denegar a un usuario sobre un archivo?

- ninguno
- conocimiento
- ejecución
- lectura
- adición
- actualización
- cambio de protección
- borrado (página 528)

12.10. Enumere y defina brevemente tres métodos de agrupamiento

- **Bloques fijos: se** usan registros de longitud fija, guardándose en cada bloque un número entero de registros. Puede existir espacio sin usar al final de cada bloque. Esto se denomina fragmentación interna.
- **Bloques de longitud variable con tramos:** se usan registros de longitud variable que se agrupan en bloques sin dejar espacio sin usar. De este modo, algunos registros deben abarcar dos bloques, indicando el tramo de continuación con un puntero al bloque siguiente.
- **Bloques de longitud variable sin tramos:** se usan registros de longitud variable, pero no se dividen en tramos. En la mayoría de los bloques habrá un espacio desperdiciado, debido a la imposibilidad de aprovechar el resto del bloque si el registro siguiente es mayor que el espacio sin usar restante. (Página 530)

12.11. Enumere y defina brevemente tres métodos de asignación de archivos

- Asignación contigua: cuando se crea una archivo se crea un único conjunto contiguo de bloques, los bloques están juntos en el disco.
- Asignación encadenada: cada bloque del archivo contiene un puntero al siguiente bloque, los bloques pueden estar dispersos en el disco.
- Asignación indexada: hay un índice que contiene las direcciones de los bloques del archivo, los bloques pueden estar dispersos en el disco. (ver páginas 533,534 y 535)

Nota: importante en este capitulo "I-Nodos"

Capítulo 13 -	Proceso distribuido, cliente/servidor y agrupaciones

Resumen:

El proceso cliente/servidor es la clave para comprender el potencial de los sistemas de información y las redes para incrementar significativamente la productividad de las organizaciones. Con la ejecución cliente/servidor, las aplicaciones se distribuyen a usuarios en estaciones de trabajo monousuario y computadores personales. Al mismo tiempo que los recursos que se pueden y deben compartir se mantienen en los sistemas servidor, están disponibles para todos los clientes. De esta forma, la arquitectura cliente/servidor es una mezcla de ejecución centralizada y descentralizada.

Normalmente, el sistema cliente ofrece una interfaz gráfica de usuario (GUI) que permite al usuario sacar provecho de múltiples aplicaciones con un aprendizaje mínimo y con relativa facilidad. Los servidores dan soporte a utilidades compartidas, como los sistemas de gestión de bases de datos. La aplicación real se divide entre el cliente y el servidor como una forma de intentar optimizar la facilidad de uso y el rendimiento.

El mecanismo clave necesario en cualquier sistema distribuido es la comunicación entre procesos. Generalmente se utilizan dos técnicas. Un servicio de paso de mensajes generaliza el paso de mensajes de un sistema único. Se aplican las mismas clases de convenciones y de normas de sincronización. Otro método es la utilización de la llamada a procedimiento remoto. Esta es una técnica por la que dos programas de diferentes máquinas interactúan utilizando la sintaxis y semántica de llamada/retorno a procedimiento. Tanto los programas llamados como los que llaman se comportan como si el programa asociado se estuviera ejecutando en la misma máquina.

Una agrupación es un grupo de computadores completos interconectados trabajando juntos como un recurso de ejecución unificado que puede crear la ilusión de ser una sola máquina. El término *computador completo* se refiere a un sistema que puede funcionar por sí solo, por separado de la agrupación.

Cuestiones de Repaso:

13.1. ¿Qué es el proceso cliente/servidor.?

Al igual que otras tendencias dentro del campo de la informática, el proceso cliente/servidor llega con su propia jerga de palabras. La Tabla 13.1 cita algunos de los términos que se encuentran generalmente en las descripciones de las aplicaciones y productos cliente/servidor.

La Figura 13.1 intenta resumir lo esencial de los conceptos cliente/servidor. Como el término sugiere. un entorno cliente/servidor está poblado de clientes y servidores. Las máquinas cliente son, en general, PC monousuario o puestos de trabajo que ofrecen una interfaz muy amigable para el usuario final. Los puestos de cliente presentan, en general, un tipo de interfaz gráfica que es más cómoda para los usuarios, incluyendo el uso de ventanas y un ratón. Algunos ejemplos comunes de este tipo de interfaces son las ofrecidas por Microsoft Windows y Macintosh. Las aplicaciones de clientes están confeccionadas para ser fáciles de usar e incluyen herramientas tan familiares como puedan ser las hojas de cálculo.

Tabla 13.1. Terminología cliente/servidor.

Middleware Un conjunto de controladores, API u otro software que mejora la conectividad entre las aplicaciones de cliente y un servidor.
Base de datos relacional Una base de datos en donde el acceso a la información está limitado por la selección de filas que satisfacen todos los criterios de búsqueda.
Servidor Un computador, generalmente una estación de trabajo muy potente, un minicomputador o un main *frame*, que contiene información para que los clientes de red puedan manipularía.
Lenguaje de consulta estructurado (SQL, ***Structured Ouery Language)*** Un lenguaje desarrollado por IBM y estandarizado por ANSI para direccionar, crear, actualizar o consultar bases de datos relacionales.

(Paginas 557 y 558)

13.2. ¿Qué diferencia al proceso cliente, servidor de cualquier otro tipo de proceso de datos distribuido?

- Se depositan aplicaciones amigables en los sistemas de usuario
- Se dispersan las aplicaciones y se centralizan las bases de datos corporativas
- Existe un compromiso tendiente a sistemas abiertos y modulares.
- El trabajo en red es fundamental

(Páginas 558 y 559)

13.3. ¿Cuál es el papel de una arquitectura de comunicaciones como TCP/IP en un entorno cliente/servidor?

Es el protocolo de comunicaciones más comúnmente usado para interoperar entre cliente y servidor.

13.4. Estudie las razones para ubicar aplicaciones en el cliente, en el servidor o para hacer una división entre cliente y servidor.
En el mejor de los casos, las funciones reales de la aplicación pueden repartirse entre cliente y servidor de forma que se optimen los recursos de la red y de la plataforma, así como la capacidad de los usuarios para realizar varias tareas y cooperar el uno con el otro en el uso de recursos compartidos. En algunos casos, estos requisitos dictan que el grueso del software de la aplicación se ejecute en el servidor, mientras que, en otros casos, la mayor parte de la lógica de la aplicación se ubica en el cliente. (página 560)

13.5. ¿Qué son clientes gruesos y delgados y cuáles son las diferencias filosóficas de los dos métodos'?
- Cliente Grueso: máquina conectada a un servidor pero que posee una capacidad importante de procesamiento local.
- Cliente delgado: máquina conectada a una servidor potente que realiza pocos o ningún computo.

La diferencia filosófica más importante tiene que ver con la centralización de datos y aplicaciones.

13.6. Sugiera los pros y los contras de las estrategias del cliente grueso y del delgado.
(ver páginas 562, 563 y 564)

13.7. Sugiera las razones ocultas de la arquitectura cliente/servidor de tres capas.
- mejor control de los datos entrantes y salientes
- posibilidad de aplicar mejor seguridad
- posibilidad de escalabilidad sin afectar las interfases
- ...

13.8. ¿Qué es el middleware'?
Para alcanzar las ventajas reales de la filosofía cliente$_1$servidor, los desarrolladores deben disponer de un conjunto de herramientas que proporcionen una manera uniforme de acceder a los recursos del sistema en todas las plataformas. Esto servirá para que los programadores construyan aplicaciones que no sólo parezcan las mismas en PC y puestos de trabajo diferentes, sino que también utilicen el mismo método de acceso a los datos, sin importar la ubicación de los mismos.
La forma más común de cumplir con este requisito es utilizar interfaces estándares de programación y protocolos que se sitúen entre la aplicación y el software de comunicaciones y el sistema operativo. Dichas interfaces y protocolos estándares han venido a llamarse *,middleware (página 566)*.

13.9. Puesto que existen estándares como TCP/IP, ¿por qué es necesario el middleware'?
TCP/IP es una interfaz para el software de comunicaciones y el middleware es una interfaz entre aplicaciones.

13.10. Enumere algunas ventajas e inconvenientes de las primitivas bloqueantes y no bloqueantes en el paso de mensajes.
(...)Con primitivas no bloqueantes. o asíncronas, un proceso no queda suspendido como resultado de un ***send*** o un ***receive.*** De esta forma, cuando un proceso emita una primitiva *Send,* el sistema operativo le devolverá el control tan pronto como el mensaje se haya puesto en cola para su transmisión o se haya hecho una copia. Si no se hace copia, cualquier cambio que realice el emisor en el mensaje antes de la transmisión o durante la misma, se hará bajo la responsabilidad del mismo. Cuando el mensaje se haya transmitido o se haya copiado a un lugar seguro para su posterior transmisión. se interrumpe al proceso emisor para informarle de que el buffer del mensaje puede reciclarse. De forma similar. un *receive* no bloqueante lo emite un proceso para después seguir ejecutándose. Cuando llega un mensaje, se informa al proceso mediante interrupción o bien este puede comprobar periódicamente su estado.
Las primitivas no bloqueantes ofrecen un empleo eficiente y flexible del servicio de paso de mensajes. La desventaja (le este enfoque es que los programas que emplean estas primitivas Son difíciles de probar y depurar. Las secuencias irreproducibles dependientes del tiempo pueden originar problemas sutiles y complicados. (...) (página 569)

13.11. Enumere algunas ventajas e inconvenientes del enlace persistente y no persistente en RPC.
Los **enlaces no persistentes** suponen que la conexión lógica se establece entre dos procesos en el momento de la llamada remota y que la conexión se pierde tan pronto como se devuelvan los valores. Como una conexión requiere el mantenimiento de información de estado en ambos extremos, consume recursos. El estilo no persistente se utiliza para conservar dichos recursos. Por otro lado, el coste de establecer las conexiones hace que los enlaces no persistentes no sean muy apropiados para procedimientos remotos que un mismo llamador invoca con frecuencia.
Con enlaces persistentes, una conexión establecida para una llamada a un procedimiento remoto se mantiene después de que el procedimiento termina. La conexión puede utilizarse para futuras llamadas a procedimiento remoto. Si transcurre un periodo de tiempo determinado sin actividad en la conexión, la misma finaliza. Para aplicaciones que realicen llamadas a procedimiento remoto repetidas veces, el enlace persistente mantiene la conexión lógica y permite que una secuencia de llamadas y retornos utilice la misma conexión. (Página 574)

13.12. Enumere algunas ventajas e inconvenientes de las RPC sincronas y asincronas.
Los conceptos de llamadas a procedimiento remoto síncronas y asíncronas son análogos a los de mensajes bloqueantes y no bloqueantes. Las llamadas tradicionales a procedimiento remoto son sincronas, lo que requiere que el proceso llamador espere hasta que el proceso llamado devuelva un valor. Así, la **RPC** síncrona se comporta de manera muy parecida a una llamada a subrutina.
La RPC síncrona es fácil de comprender y de programar puesto que su comportamiento es predecible. Sin embargo, no es capaz de explotar por completo el paralelismo inherente a las aplicaciones distribuidas. Esto limita el tipo de interacción que las aplicaciones distribuidas pueden realizar, que así obtienen un rendimiento menor. (página 574)

13.13. Enumere y defina brevemente cuatro métodos diferentes de agrupación.

Un método habitual, más antiguo, conocido como espera **pasiva** consiste simplemente en hacer que un computador gestione toda la carga de proceso mientras que otro computador permanece inactivo, esperando para hacerse cargo en caso de fallo de la primaria. Para coordinar las máquinas, el sistema activo o primario envía periódicamente un mensaje de «latido» a la máquina en espera. Si estos mensajes dejan de llegar, el computador en espera supone que el servidor primario ha fallado y se pone a funcionar. Este método aumenta la disponibilidad, pero no mejora el rendimiento. Es más, si la única información que se intercambian los dos sistemas es un mensaje de latido, y silos dos sistemas no comparten discos comunes, el computador en espera ofrece una funcionalidad de respaldo pero no tiene acceso a las bases de datos gestionadas por la primaria.

La espera pasiva generalmente no se denomina agrupación. El término *agrupación* se reserva para múltiples computadores interconectados que realizan todas proceso activo mientras mantienen la imagen externa de un solo sistema. Para hacer referencia a este tipo de configuración generalmente se utiliza el término **secundaria activa.** Se pueden identificar tres métodos de agrupaciones:

servidores separados, nada compartido, y memoria compartida.

En la primera aproximación a las agrupaciones, cada computador es un servidor **separado,** con sus propios discos y sin compartimiento de discos entre los sistemas (Figura 13.14a). Este esquema proporciona un alto rendimiento así como una alta disponibilidad. En este caso, se necesita algún tipo de gestión o software de planificación para asignar las solicitudes entrantes de los clientes a los servidores para que así la carga esté equilibrada y se logre una alta utilización. Sería deseable disponer de capacidad de tolerancia a los fallos, lo que significa que si un computador falla cuando está ejecutando una aplicación, otro computador de la agrupación puede hacerse cargo de la misma y terminarla. Para que esto ocurra los datos deben copiarse constantemente entre los sistemas, de forma que cada uno de los sistemas tenga acceso a los datos actuales de los otros. La sobrecarga de este intercambio de datos asegura una alta disponibilidad a costa de penalizar el rendimiento.

Para reducir la sobrecarga de comunicaciones, la mayoría de las agrupaciones se componen actualmente de servidores conectados a discos comunes (Figura 13.1 4b). Una variación de este en-foque es el que se denomina simplemente **compartir nada.** En este enfoque, los discos comunes se dividen en volúmenes y cada volumen pertenece a un único computador. Si ese computador falla, la agrupación se debe reconfigurar para asociar los volúmenes del computador que ha fallado a otros computadores.

También es posible hacer que múltiples computadores compartan los mismos discos al mismo tiempo (enfoque denominado **compartir** disco), para que cada computador tenga acceso a todos los volúmenes de todos los discos. Ente enfoque necesita del uso de algún tipo de función de bloqueo para garantizar que solo puede acceder a los datos un solo computador cada vez.

(Página 577)

Capítulo 14 -	Gestión distribuida de Procesos

Resumen:

Un sistema operativo distribuido puede ofrecer migración de procesos, es decir, la transferencia de una parte del estado de un proceso de una máquina a otra para que el proceso se ejecute en la máquina de destino. La migración de procesos puede utilizarse para equilibrar la carga, mejorar el rendimiento al disminuir la actividad de comunicación, aumentar la disponibilidad o permitir a los procesos acceder a servicios especializados remotos.

En un sistema distribuido, suele ser importante construir la información del estado global, para resolver la contienda por los recursos y coordinar los procesos. Debido al retardo variable e impredecible en la transmisión de mensajes, se debe tener cuidado en garantizar que los diferentes procesos se ponen de acuerdo en el orden en que se producen los sucesos.

La gestión de procesos de un sistema distribuido incluye servicios para hacer cumplir la exclusión mutua y para tomar acciones de tratamiento del interbloqueo. En ambos casos, los problemas son más complejos que en un sistema sencillo.

Cuestiones de Repaso:

14.1. Comente alguna de las razones para la implementación de la migración de procesos.

- compartimiento de carga
- rendimiento de las comunicaciones
- disponibilidad
- utilización de capacidades especiales

14.2. ¿Cómo se gestiona el espacio de direcciones del proceso durante la migración?

Se gestiona por:

- transferencia completa
- Copia anticipada
- Transferencia (modificada)
- Copia por referencia
- Volcado (flushing)

14.3. ¿Cuáles son las causas para la migración de procesos preferente y no preferente?

El reparto de carga (¿?)

14.4. ¿Por qué es imposible determinar un estado global cierto?

Porque los datos que se pueden obtener de un sistema remoto mediante mensajes, al momento de llegar, ya son viejos.

14.5. ¿Cuál es la diferencia entre exclusión mutua distribuida mediante un enfoque centralizado y mediante un enfoque distribuido?

Que una depende de una memoria común y otra del paso de mensajes.

14.6. Defina los dos tipos de interbloqueo distribuido.

Capítulo 15 -	Seguridad

Resumen:

Los requisitos de seguridad se evalúan mejor examinando las diversas amenazas a la seguridad a las que se enfrenta una organización. La interrupción del servicio es una amenaza a la disponibilidad. La interceptación de información es una amenaza al secreto. Por último, la modificación de información legítima y la invención no autorizada de información son amenazas a la integridad.

Un campo clave de la seguridad en los computadores es la protección de la memoria. Esta es esencial en cualquier sistema en el que haya varios procesos activos al mismo tiempo. Los esquemas de memoria virtual suelen estar equipados con los mecanismos apropiados para esta tarea.

Otra técnica importante de seguridad es el control de acceso. La finalidad del control de acceso es asegurarse que sólo los usuarios autorizados tienen acceso a un sistema particular y a sus recursos, así como asegurar que el acceso y la modificación de partes concretas de los datos están restringidas a los individuos y programas autorizados. Estrictamente hablando, el control de acceso es un problema de seguridad de computadores más que de seguridad de redes. Es decir, en la mayoría

Cuestiones de Repaso:

15.1. ¿Cuáles son los requisitos fundamentales que aborda la seguridad de computadores?

- Secreto
- Integridad
- Disponibilidad
- Autenticidad

15.2. ¿Cuál es la diferencia entre amenaza a la seguridad activa y pasiva?

Amenazas activas: son las del tipo de las escuchas a escondidas o control de trasmisiones.

Amenazas pasivas: suponen alteraciones del flujo de datos o creación de algún flujo falso.

15.3. Enumere y defina brevemente las categorías de amenazas a la seguridad activa y pasiva.

Pasivas:

- revelación del contenido del mensaje
- análisis de tráfico

Activas:

- suplantación
- repetición
- modificación de mensajes
- privación de servicio

15.4. ¿Qué elementos se necesitan en las técnicas de control de acceso más comunes?

Procesadores, memoria, Dispositivos de E/S, programasy Datos

15.5. En el control de acceso, ¿cuál es la diferencia entre un sujeto y un objeto?

Un sujeto es una entidad capaz de acceder a los objetos y los objetos son los elementos que deben controlarse.

15.6. Explique el propósito de la semilla de la Figura *15.5.*

Es un valor aleatorio que modifica el resultado del algoritmo.

15.7. Explique la diferencia entre la detección de intrusiones por anomalías estadísticas y la detección de intrusiones basada en reglas.

Uno intenta detectar intrusiones por alejamiento del comportamiento normal y otro por alejamiento del comportamiento correcto.

15.8. A los programas malignos por correo electrónico con archivos adjuntos o por VBS desarrollados en 1999 y 2000 (por ejemplo, Melissa o Love letter) en los medios de comunicación se les llama *virus de correo electrónico*. ¿Podría ser más exacto el término *gusanos de correo electrónico?*

Si, porque utiliza los servicios de correo electrónico, la capacidad de ejecución remota y la capacidad de conexión remota.

15.9. ¿Qué papel interpreta el cifrado en el diseño de virus?

Permite la mutación de los virus polimorfos

15.10. ¿Cuáles son las dos aproximaciones generales para atacar un esquema de cifrado clásico?

Se utiliza tecnología de descifrado genérico y la inmunización digital.

15.11. ¿Qué es el DES y el DEA triple?

Son técnicas de cifrado de datos estándar.

15.12. ¿Cómo se espera que el AES sea una mejora sobre el DEA triple'?

AES es más eficiente y soporta claves de 128,, 198 y 256 bits.

15.13. ¿Qué criterio de evaluación se usará para calcular los candidatos AES?

Seguridad, eficiencia, necesidades de memoria, adaptación al software y hardware y flexibilidad.

15.14. Explique la diferencia entre el cifrado clásico y el cifrado por clave pública.
Uno se basa en funciones matemáticas y otro en operaciones sobre patrones de bits.

15.15. ¿Cuáles son las diferencias entre los términos *clave pública, clave privada, clave secreta*?
La clave que se utiliza en el cifrado clásico se llama clave secreta. Las dos claves que utilizadas para el cifrado con clave pública se conocen como clave publica y privada,. Sin excepción la clave privada se debe mantener en secreto, pero se denomina clave privada para evitar confusiones (página 681)