

Universidad
Industrial de
Santander



Taller_05

Presentado por:

Manuel De Angel

Cod. De estudiante 2192510

Presentado a:

Pablo J. Rojas

Sistemas Operacionales

Cod. De asignatura 22972

22/04/2023

1. Compare las diferencias entre la planeación a corto plazo y largo plazo.

La **planeación a corto plazo** se enfoca en la asignación de recursos y la toma de decisiones inmediatas para asegurar el buen funcionamiento del sistema operativo en el día a día. Esto incluye tareas como la asignación de procesos, la gestión de memoria, la gestión de entradas y salidas, y la planificación de tareas.

La **planeación a largo plazo** se enfoca en la planificación estratégica a largo plazo, con el objetivo de mejorar la eficiencia y la eficacia del sistema operativo en el futuro. Esta planeación incluye la determinación de objetivos a largo plazo, la definición de estrategias para alcanzar esos objetivos, la asignación de recursos para proyectos de desarrollo y la evaluación de las tendencias tecnológicas.

2. Caracterice dos procesos que se puedan considerar a mediano plazo

1.1 **Definición de objetivos y metas a mediano plazo:** En esta etapa, se identifican los objetivos y metas específicas que se deben alcanzar en un período de tiempo que oscila entre uno y tres años. Estos objetivos deben ser realistas, específicos, medibles y relevantes para el éxito del sistema operativo.

2.1 **Evaluación de recursos:** En esta etapa, se evalúan los recursos necesarios para alcanzar los objetivos a mediano plazo. Esto incluye la evaluación de la infraestructura existente, los recursos humanos, los recursos financieros y los recursos tecnológicos necesarios para la implementación efectiva de los objetivos.

3. Describa las acciones que toma el kernel para el cambio de contexto entre procesos.

- *Salvado del contexto actual:* Antes de cambiar de proceso, el kernel debe guardar el contexto actual del proceso que se encuentra en ejecución. Esto incluye la información de los registros del procesador, los punteros a las estructuras de datos del proceso y otros elementos relevantes del estado del proceso.
- *Selección del proceso siguiente:* Después de guardar el contexto del proceso actual, el kernel debe seleccionar el proceso siguiente que recibirá el control del procesador. Esto puede hacerse de diferentes maneras, dependiendo del algoritmo de planificación utilizado por el kernel.
- *Carga del contexto del proceso siguiente:* Una vez seleccionado el proceso siguiente, el kernel debe cargar el contexto del proceso en la CPU. Esto incluye la recuperación de los registros del procesador, los punteros a las estructuras de datos del proceso y otros elementos relevantes del estado del proceso.
- *Inicio de la ejecución del proceso siguiente:* Después de cargar el contexto del proceso siguiente, el kernel debe iniciar la ejecución del proceso seleccionado, transfiriendo el control del procesador al primer punto de ejecución del proceso.

4. Enumere y describa cinco actividades de un OS enfocadas a la administración de archivos

Comunicación síncrona:

Ventajas:

Es más fácil de implementar y depurar debido a que el proceso emisor espera la respuesta del proceso receptor antes de continuar su ejecución.

Es útil en situaciones en las que se necesita una respuesta inmediata del proceso receptor antes de continuar.

Evita problemas de concurrencia y sincronización, ya que el proceso emisor y receptor están sincronizados.

Desventajas:

Puede generar bloqueos en la aplicación si el proceso receptor no responde o se queda inactivo.

El proceso emisor debe esperar la respuesta del proceso receptor, lo que puede reducir la eficiencia y el rendimiento de la aplicación.

No es adecuada para situaciones en las que la comunicación debe ser asíncrona o para sistemas distribuidos en los que la latencia de la red puede afectar el rendimiento.

Comunicación asíncrona:

Ventajas:

Permite la comunicación asíncrona, lo que significa que el proceso emisor no tiene que esperar la respuesta del proceso receptor antes de continuar su ejecución.

Es útil en situaciones en las que el proceso emisor no necesita una respuesta inmediata del proceso receptor.

Puede aumentar la eficiencia y el rendimiento de la aplicación, ya que el proceso emisor no tiene que esperar la respuesta del proceso receptor.

Desventajas:

Puede ser más difícil de implementar y depurar debido a que el proceso emisor y receptor pueden estar ejecutándose de forma independiente.

Puede generar problemas de concurrencia y sincronización si los procesos emisor y receptor no están correctamente sincronizados.

Puede ser difícil de utilizar en sistemas distribuidos en los que la latencia de la red puede afectar el rendimiento.

5. Defina las ventajas y desventajas desde el punto de vista del OS para envío por copia y envío por referencia.

1.1 Envío por copia:

Ventajas:

Es más seguro y confiable, ya que el valor del argumento pasado a la función no puede ser modificado por la función.

Evita problemas de sincronización y concurrencia, ya que cada función tiene su propia copia del argumento.

Es más fácil de implementar y depurar.

Desventajas:

Puede ser menos eficiente en términos de uso de memoria, ya que se deben realizar copias adicionales de los datos para cada función que se llama.

2.1 Envío por referencia:

Ventajas:

Es más eficiente en términos de uso de memoria, ya que no se requiere realizar copias adicionales de los datos.

Es más flexible, ya que la función puede modificar directamente el valor del argumento y devolver un resultado.

Desventajas:

Es menos seguro y confiable, ya que la función puede modificar directamente el valor del argumento.

Puede generar problemas de sincronización y concurrencia si varios procesos o threads comparten la misma referencia.

Puede ser más difícil de implementar y depurar debido a que el valor del argumento puede ser modificado por la función.

6. Defina las ventajas y desventajas desde el punto de vista del OS para mensajes de tamaño fijo y de tamaño variable.

Mensajes de tamaño fijo:

Ventajas:

Son más fáciles de implementar y depurar, ya que el sistema operativo sabe de antemano el tamaño exacto del mensaje.

Son más eficientes en términos de uso de memoria, ya que se pueden reservar bloques de memoria de tamaño fijo para almacenar los mensajes.

Son menos propensos a errores, ya que los procesos emisor y receptor tienen una comprensión clara de la estructura y tamaño del mensaje.

Desventajas:

Pueden ser menos flexibles, ya que el tamaño fijo del mensaje puede limitar la cantidad y el tipo de datos que se pueden enviar.

Pueden generar desperdicio de memoria si el mensaje no utiliza todo el espacio reservado para él.

Mensajes de tamaño variable:

Ventajas:

Son más flexibles, ya que pueden adaptarse a diferentes tipos y cantidades de datos.

Son menos propensos a generar desperdicio de memoria, ya que sólo utilizan la cantidad necesaria de memoria para almacenar los datos.

Desventajas:

Pueden ser más difíciles de implementar y depurar, ya que el sistema operativo necesita gestionar el tamaño dinámico del mensaje y los posibles errores asociados.

Pueden ser menos eficientes en términos de uso de memoria, ya que el sistema operativo necesita buscar y asignar bloques de memoria de tamaño variable para almacenar los mensajes.

7. Describa los estados de un proceso.

- 1.1 **Nuevo:** En este estado, el proceso se está creando. Es decir, se han inicializado todas las estructuras de datos necesarias y se han reservado los recursos necesarios para su ejecución, pero aún no se han asignado los recursos de hardware, como la CPU y la memoria principal. En este estado, el proceso no está listo para su ejecución.
- 2.1 **En ejecución:** Una vez que se han asignado los recursos de hardware, el proceso pasa al estado de "En ejecución". En este estado, el proceso está ejecutando sus instrucciones y utilizando los recursos asignados, como la CPU y la memoria principal. El proceso permanecerá en este estado hasta que termine su ejecución o se le interrumpa por algún motivo, como la llegada de una interrupción o la expiración de un temporizador.
- 3.1 **En espera:** Cuando un proceso necesita esperar a que ocurra un evento, como la llegada de una entrada/salida, se coloca en el estado de "En espera". En este estado, el proceso no está utilizando los recursos de la CPU y está esperando a que se produzca el evento necesario para continuar su ejecución. Una vez que se produce el evento, el proceso puede volver al estado de "En ejecución".
- 4.1 **Preparado:** Cuando un proceso está listo para su ejecución pero aún no se le ha asignado la CPU, se coloca en el estado de "Preparado". En este estado, el proceso está esperando a que la CPU esté disponible para su asignación. Una vez que la CPU está disponible, el proceso pasa al estado de "En ejecución" y comienza su ejecución.
- 5.1 **Terminado:** Cuando un proceso ha terminado su ejecución, se coloca en el estado de "Terminado". En este estado, el proceso ha liberado todos los recursos que estaba utilizando y ha notificado al sistema operativo que su ejecución ha finalizado. En este estado, el proceso permanecerá hasta que el sistema operativo limpie todas las estructuras de datos asociadas con el proceso y lo elimine de la lista de procesos en ejecución.

8. Que datos se encuentran en un PCB

- **Identificador de proceso:** es un número único que identifica de manera unívoca a cada proceso en el sistema operativo.
- **Estado del proceso:** indica en qué estado se encuentra el proceso (nuevo, en ejecución, en espera, preparado o terminado).

- **Contador de programa (PC):** indica la dirección de memoria donde se encuentra la próxima instrucción que debe ejecutar el proceso.
- **Registros del procesador:** almacenan los valores de los registros del procesador (por ejemplo, registros de propósito general, punteros de pila, etc.) utilizados por el proceso.
- **Información de planificación:** incluye la prioridad del proceso, el tiempo de CPU utilizado, el tiempo de espera, etc.
- **Información de memoria:** indica la ubicación de la memoria física asignada al proceso y la dirección virtual a la que se ha mapeado.
- **Información de recursos:** incluye información sobre los recursos del sistema que el proceso ha solicitado o adquirido, como archivos abiertos, dispositivos de entrada/salida, etc.
- **Información de estado de entrada/salida:** indica si el proceso está esperando a que se complete alguna operación de entrada/salida y, si es así, qué dispositivo de entrada/salida está utilizando.
- **Información de seguridad:** incluye información de seguridad relacionada con el proceso, como los permisos de usuario y los límites de recursos del sistema.

9. Describa un modelo de comunicación Cliente-Servidor

El cliente es una aplicación que solicita servicios o recursos al servidor, mientras que el servidor es una aplicación que proporciona servicios o recursos a los clientes. La comunicación entre el cliente y el servidor se realiza mediante protocolos de comunicación estándar, como HTTP, TCP/IP, entre otros.

En este modelo, el cliente y el servidor pueden estar ubicados en diferentes máquinas o dispositivos, y la comunicación entre ellos se realiza a través de la red. El cliente envía una solicitud al servidor, que procesa la solicitud y envía una respuesta al cliente.

La interacción entre el cliente y el servidor sigue un patrón establecido. El cliente envía una solicitud al servidor, que la procesa y devuelve una respuesta al cliente. El cliente, a su vez, procesa la respuesta y puede enviar otra solicitud al servidor, y así sucesivamente.

El modelo Cliente-Servidor tiene varias ventajas, como la escalabilidad, la flexibilidad y la capacidad de distribución. Los servidores pueden ser escalados para manejar grandes cargas de trabajo y múltiples clientes, y los clientes pueden acceder a los servicios del servidor desde cualquier lugar de la red. Además, este modelo permite una mayor flexibilidad en la distribución de recursos y servicios, ya que los clientes pueden acceder a los recursos y servicios del servidor de forma remota.