

INVESTIGACION 04

PRESENTADO POR:

SEBASTIÁN DAVID MENDOZA ALVARADO

PRESENTADO A:

PABLO ROJAS

Docente

UNIVERSIDAD INDUSTRIAL DE SANTANDER

FACULTAD DE FISICOMECHANICAS

ESCUELA DE INGENIERIA DE SISTEMAS

SISTEMAS OPERACIONALES

BUCARAMANGA

2023

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

A continuación, se presenta un cuadro comparativo que resume las principales diferencias entre la planeación a corto plazo y la planeación a largo plazo:

Características	Planeación a corto plazo	Planeación a largo plazo
Alcance temporal	Corto plazo (días a meses)	Largo plazo (años a décadas)
Foco	Objetivos y acciones a corto plazo para cumplir con metas a largo plazo	Metas y objetivos a largo plazo y estrategias para alcanzarlos
Flexibilidad	Mayor flexibilidad para ajustarse a cambios inmediatos en el entorno empresarial	Menos flexible, requiere un enfoque estructurado para asegurarse de cumplir los objetivos a largo plazo
Nivel de detalle	Más detallada y específica	Más general y estratégica
Ejemplos de objetivos	Incrementar ventas este trimestre	Expandir el negocio a nuevos mercados internacionales en los próximos 5 años

En resumen, mientras que la planeación a corto plazo se enfoca en objetivos y acciones específicas que deben ser realizadas en el corto plazo para cumplir con las metas a largo plazo, la planeación a largo plazo se enfoca en metas y objetivos a largo plazo y en las estrategias para alcanzarlos. Además, la planeación a corto plazo es más flexible y detallada, mientras que la planeación a largo plazo es menos flexible y más general en su enfoque.

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

Existen varios procesos que se pueden considerar a mediano plazo, pero a continuación se describen dos de ellos:

Planificación estratégica: La planificación estratégica es un proceso a mediano plazo que implica la identificación de objetivos a largo plazo y la determinación de las acciones y recursos necesarios para alcanzarlos. Este proceso suele tener un horizonte temporal de 3 a 5 años y requiere la participación activa de los líderes y gerentes de la organización. La planificación estratégica implica la evaluación del entorno empresarial, la identificación de oportunidades y amenazas, la evaluación de los recursos y capacidades de la organización, la definición de objetivos claros y realistas, y la identificación de los pasos específicos que se deben tomar para lograr los objetivos.

Gestión de proyectos: La gestión de proyectos es un proceso a mediano plazo

que implica la planificación, organización, coordinación y seguimiento de actividades específicas para lograr un objetivo específico en un plazo determinado. Este proceso suele tener un horizonte temporal de varios meses a un par de años y requiere la participación activa de un equipo de trabajo dedicado. La gestión de proyectos implica la identificación de objetivos y entregables específicos, la asignación de tareas y responsabilidades, la gestión de recursos y presupuestos, la identificación de riesgos y la implementación de medidas de mitigación, y el monitoreo y seguimiento continuo del progreso del proyecto para asegurar que se alcancen los objetivos establecidos en el plazo determinado.

En resumen, tanto la planificación estratégica como la gestión de proyectos son procesos a mediano plazo que requieren la identificación de objetivos claros y la implementación de acciones específicas para lograrlos. Ambos procesos implican una planificación cuidadosa y la asignación de recursos y responsabilidades para alcanzar los objetivos establecidos en un plazo determinado.

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

El cambio de contexto entre procesos es un proceso fundamental en la programación de sistemas operativos multitarea. Cuando el sistema operativo se ejecuta en modo kernel, es responsable de administrar los recursos del sistema y asignar tiempo de CPU a cada proceso en ejecución. A continuación, se describen las acciones que toma el kernel para el cambio de contexto entre procesos:

- A) **Interrupción de temporizador:** El kernel utiliza una interrupción de temporizador para programar cambios de contexto periódicos. Cuando un temporizador interno del sistema operativo llega a cero, se genera una interrupción que indica al kernel que debe seleccionar el próximo proceso para ejecutar.
- B) **Guardado del contexto actual:** El kernel guarda el contexto actual del proceso en ejecución, que incluye el contenido de los registros de la CPU, la información de estado y la pila de llamadas. El contexto del proceso se almacena en la memoria del sistema para que pueda ser restaurado más tarde cuando el proceso vuelva a ejecutarse.
- C) **Selección del próximo proceso:** El kernel selecciona el próximo proceso a ejecutar utilizando un algoritmo de planificación de procesos. El algoritmo de planificación determina qué proceso debe ejecutarse a continuación, basándose en factores como el tiempo de ejecución del proceso, su prioridad y su estado actual.
- D) **Restauración del contexto del próximo proceso:** Una vez que el kernel ha seleccionado el próximo proceso a ejecutar, se restaura su contexto desde la memoria del sistema. El contenido de los registros de la CPU, la información de estado y la pila de llamadas se cargan desde la memoria a la CPU, de manera que el proceso pueda continuar su ejecución desde el punto en que se detuvo.
- E) **Cambio a modo usuario:** Finalmente, el kernel cambia de nuevo al modo usuario y permite que el proceso recién seleccionado continúe su ejecución normalmente.

En resumen, el cambio de contexto entre procesos implica la interrupción periódica del temporizador, la selección del próximo proceso a ejecutar mediante un algoritmo de planificación, y la restauración del contexto del proceso seleccionado para que pueda continuar su ejecución desde donde se detuvo. El kernel del sistema operativo es responsable de realizar estas acciones para permitir una multitarea eficiente y equitativa.

4) Defina las ventajas y desventajas desde el punto de vista del programador para comunicación síncrona y asíncrona.

La comunicación entre procesos es una técnica importante en la programación de sistemas operativos y aplicaciones distribuidas. La comunicación síncrona y asíncrona son dos métodos diferentes para la comunicación entre procesos, y cada uno tiene ventajas y desventajas desde el punto de vista del programador. A continuación, se describen las ventajas y desventajas de cada uno:

Comunicación síncrona:

Ventajas:

- La comunicación síncrona es fácil de implementar y entender, ya que los procesos esperan activamente una respuesta del otro proceso antes de continuar con la ejecución.
- La comunicación síncrona proporciona una sincronización natural entre los procesos, lo que puede simplificar el diseño de la aplicación.
- La comunicación síncrona es útil para situaciones en las que un proceso necesita datos o recursos específicos de otro proceso antes de continuar.

Desventajas:

- La comunicación síncrona puede ser ineficiente en situaciones en las que un proceso necesita esperar una respuesta de otro proceso antes de continuar con la ejecución.
- La comunicación síncrona puede ser propensa a bloqueos si uno de los procesos se queda esperando una respuesta que nunca llega.

Comunicación asíncrona:

Ventajas:

- La comunicación asíncrona puede ser más eficiente que la comunicación síncrona, ya que los procesos pueden continuar ejecutándose mientras esperan la respuesta de otro proceso.
- La comunicación asíncrona es menos propensa a bloqueos que la comunicación síncrona, ya que los procesos no tienen que esperar activamente una respuesta.
- La comunicación asíncrona es útil en situaciones en las que un proceso no necesita una

respuesta inmediata de otro proceso para continuar con la ejecución.

Desventajas:

- La comunicación asíncrona puede ser más difícil de implementar y entender que la comunicación síncrona, ya que los procesos deben manejar las respuestas en un momento posterior de la ejecución.
- La comunicación asíncrona puede ser propensa a errores si los procesos no se sincronizan adecuadamente, lo que puede resultar en datos inconsistentes o comportamientos impredecibles.

En resumen, la elección entre comunicación síncrona y asíncrona depende de las necesidades de la aplicación y del contexto de la comunicación. La comunicación síncrona es fácil de entender y puede ser útil en situaciones en las que se necesita una respuesta inmediata, mientras que la comunicación asíncrona puede ser más eficiente y menos propensa a bloqueos en situaciones en las que los procesos no necesitan esperar activamente una respuesta.

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

El envío por copia y el envío por referencia son dos métodos diferentes para pasar parámetros a una función en un sistema operativo. Cada uno tiene ventajas y desventajas desde el punto de vista del sistema operativo. A continuación, se describen las ventajas y desventajas de cada uno:

Envío por copia:

Ventajas:

- El envío por copia garantiza que la función no modifique el valor original del parámetro, lo que puede ser útil en situaciones en las que se necesita mantener el valor original sin cambios.
- El envío por copia es fácil de implementar y entender, ya que cada función tiene su propia copia del parámetro, que se almacena en su propia pila de llamadas.
- El envío por copia es útil en situaciones en las que el parámetro es pequeño y no se espera que cambie su valor durante la ejecución de la función.

Desventajas:

- El envío por copia puede ser ineficiente en situaciones en las que el

parámetro es grande, ya que se debe copiar todo el valor a la pila de llamadas de la función, lo que puede consumir recursos de memoria y de CPU.

- El envío por copia no permite que la función modifique el valor original del parámetro, lo que puede ser un inconveniente en situaciones en las que se desea actualizar el valor original.

Envío por referencia:

Ventajas:

- El envío por referencia es más eficiente que el envío por copia, ya que solo se pasa una referencia al valor original, en lugar de copiar todo el valor.
- El envío por referencia permite que la función modifique el valor original del parámetro, lo que puede ser útil en situaciones en las que se desea actualizar el valor original.
- El envío por referencia es útil en situaciones en las que el parámetro es grande y se desea evitar la copia completa del valor.

Desventajas:

- El envío por referencia puede ser más difícil de entender y de implementar que el envío por copia, ya que se deben manejar las referencias y los punteros a la memoria.
- El envío por referencia puede ser propenso a errores si no se manejan adecuadamente las referencias y los punteros a la memoria, lo que puede resultar en datos inconsistentes o comportamientos impredecibles.

En resumen, la elección entre envío por copia y envío por referencia depende de las necesidades de la función y del contexto de la llamada. El envío por copia es útil en situaciones en las que se necesita mantener el valor original sin cambios y el parámetro es pequeño, mientras que el envío por referencia es más eficiente y permite que la función modifique el valor original del parámetro, pero puede ser más propenso a errores si no se manejan adecuadamente las referencias y los punteros a la memoria.

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

Los mensajes de tamaño fijo y de tamaño variable son dos formas diferentes de implementar la comunicación entre procesos en un sistema operativo. Cada uno tiene ventajas y desventajas desde el punto de vista del sistema operativo. A continuación, se describen las ventajas y desventajas de cada uno:

Mensajes de tamaño fijo:

Ventajas:

- Los mensajes de tamaño fijo son más eficientes que los mensajes de tamaño variable, ya que se pueden implementar con una estructura de datos simple y sin necesidad de realizar cálculos adicionales para manejar el tamaño del mensaje.
- Los mensajes de tamaño fijo son más predecibles que los mensajes de tamaño variable, ya que siempre se sabe el tamaño exacto del mensaje que se va a enviar o recibir.

Desventajas:

- Los mensajes de tamaño fijo pueden ser limitantes en situaciones en las que se necesita enviar un mensaje que excede el tamaño fijo especificado, lo que puede requerir la división del mensaje en partes más pequeñas o la implementación de un protocolo adicional para manejar mensajes más grandes.
- Los mensajes de tamaño fijo pueden desperdiciar espacio en la memoria si se utilizan para enviar mensajes que son más pequeños que el tamaño fijo especificado.

Mensajes de tamaño variable:

Ventajas:

- Los mensajes de tamaño variable son más flexibles que los mensajes de tamaño fijo, ya que permiten enviar mensajes de cualquier tamaño.
- Los mensajes de tamaño variable pueden ser más eficientes que los mensajes de tamaño fijo en situaciones en las que se envían mensajes de diferentes tamaños, ya que no se desperdicia espacio en la memoria para mensajes más pequeños.

Desventajas:

- Los mensajes de tamaño variable pueden ser más complejos de implementar que los mensajes de tamaño fijo, ya que se debe manejar el tamaño variable del mensaje y garantizar que se reciba el mensaje completo.
- Los mensajes de tamaño variable pueden ser menos predecibles que los

mensajes de tamaño fijo, ya que se debe realizar un cálculo adicional para determinar el tamaño del mensaje antes de enviarlo o recibirlo.

En resumen, la elección entre mensajes de tamaño fijo y de tamaño variable depende de las necesidades de la comunicación entre procesos y del contexto de la aplicación. Los mensajes de tamaño fijo son más eficientes y predecibles, pero pueden ser limitantes en situaciones en las que se necesita enviar mensajes más grandes. Los mensajes de tamaño variable son más flexibles y pueden ser más eficientes en situaciones en las que se envían mensajes de diferentes tamaños, pero pueden ser más complejos de implementar y menos predecibles.

Principio del formulario

7) Describa los estados de un proceso.

En los sistemas operativos, un proceso es una instancia de un programa en ejecución. Los procesos tienen diferentes estados a lo largo de su ciclo de vida. A continuación, se describen los estados de un proceso:

- **Nuevo:** en este estado, el proceso ha sido creado pero aún no se ha iniciado su ejecución.
- **Listo:** en este estado, el proceso está listo para ejecutarse, pero aún no se le ha asignado un tiempo de CPU. Esto puede ocurrir porque otros procesos tienen prioridad o porque el sistema operativo está esperando que se complete alguna tarea previa.
- **En ejecución:** en este estado, el proceso está siendo ejecutado en el procesador y utiliza los recursos del sistema.
- **Bloqueado:** en este estado, el proceso se encuentra temporalmente detenido debido a que está esperando a que se complete alguna tarea o evento externo, como la lectura de un archivo o la recepción de una señal.
- **Terminado:** en este estado, el proceso ha finalizado su ejecución y ha liberado los recursos que había utilizado.

Además de estos estados, hay otros dos que se pueden considerar:

- **Zombie:** este estado se refiere a un proceso que ha finalizado su ejecución, pero aún no ha sido eliminado por completo del sistema. El proceso zombie permanece en el sistema hasta que su salida es recogida por otro proceso
- **Suspensión:** este estado se refiere a un proceso que ha sido temporalmente suspendido, por ejemplo, para liberar recursos o para permitir la ejecución de otros procesos de alta prioridad. En este estado, el proceso se encuentra inactivo y no utiliza recursos del sistema.

Estos estados son comunes a la mayoría de los sistemas operativos, aunque puede haber variaciones según el sistema específico y su configuración. El conocimiento de los estados de un proceso es fundamental para entender cómo funcionan los sistemas operativos y cómo se gestiona la ejecución de los programas.

8) Que datos se encuentran en un PCB.

Un PCB (Process Control Block), también conocido como Bloque de Control de Proceso, es una estructura de datos utilizada por el sistema operativo para mantener la información de un proceso en ejecución. Los datos que se encuentran en un PCB varían según el sistema operativo específico, pero suelen incluir algunos o todos los siguientes elementos:

- **Identificación del proceso:** un identificador único para el proceso, que suele ser un número.
- **Estado del proceso:** indica en qué estado se encuentra el proceso, como listo, bloqueado, en ejecución o terminado.
- **Contador de programa:** un valor que indica el punto en el que se encuentra la ejecución del programa.
- **Información de prioridad:** información sobre la prioridad del proceso en relación con otros procesos.
- **Información de gestión de memoria:** información sobre el espacio de memoria asignado al proceso, como su dirección base y límites.
- **Información de gestión de recursos:** información sobre los recursos que el proceso ha solicitado o que le han sido asignados, como archivos abiertos, dispositivos de E/S o semáforos.
- **Información de gestión de CPU:** información sobre el uso de CPU del proceso, como el tiempo que ha estado en ejecución y el tiempo que ha estado bloqueado.
- **Información de planificación:** información utilizada por el planificador del sistema operativo para determinar qué proceso debe ejecutarse a continuación.

En general, un PCB es una estructura de datos que contiene toda la información necesaria para que el sistema operativo pueda gestionar un proceso en ejecución. Los datos específicos que se incluyen pueden variar según el sistema operativo y la configuración del sistema.

Principio del formulario

9) Describa un modelo de comunicación Cliente-Servidor.

El modelo de comunicación cliente-servidor es una arquitectura de software en la que un programa cliente solicita servicios o recursos a un programa servidor a través de una red. El modelo de cliente-servidor es uno de los más comunes en el diseño de aplicaciones de red y se utiliza en muchos tipos de sistemas distribuidos, como aplicaciones web, servicios de bases de datos, servicios de correo electrónico y muchos otros.

En este modelo, el programa cliente es el que inicia la comunicación y solicita los servicios o recursos que necesita al servidor. El servidor, por su parte, está a la espera de las solicitudes de los clientes y las procesa una por una. El servidor puede ser una aplicación independiente que se ejecuta en una computadora dedicada, o puede ser parte de una aplicación que se ejecuta en la misma máquina que el cliente.

La comunicación entre el cliente y el servidor se realiza mediante el uso de un protocolo de red específico, que establece las reglas y el formato de los mensajes que se intercambian. El protocolo puede ser TCP/IP, HTTP, FTP, SMTP, entre otros.

El modelo de comunicación cliente-servidor se divide en dos partes:

- **Cliente:** la parte del programa que inicia la comunicación y solicita los servicios o recursos al servidor.
- **Servidor:** la parte del programa que recibe las solicitudes de los clientes, procesa la solicitud y devuelve la respuesta.

El modelo de cliente-servidor ofrece varias ventajas, como una mayor eficiencia, escalabilidad, seguridad y flexibilidad. Por ejemplo, permite que múltiples clientes se conecten al mismo servidor simultáneamente, lo que mejora la eficiencia del sistema. También permite que el servidor gestione y proteja los datos, lo que mejora la seguridad.

En resumen, el modelo de comunicación cliente-servidor es una arquitectura de software en la que un programa cliente solicita servicios o recursos a un programa servidor a través de una red mediante un protocolo específico. Es un modelo comúnmente utilizado en sistemas distribuidos y proporciona una mayor eficiencia, escalabilidad, seguridad y flexibilidad.

